An alternative modelling technique for the reduction of error in decision support spreadsheets

A Thesis
submitted to the
University of Wales Institute Cardiff
For the degree of

Doctor of Philosophy

By

Simon Roy Thorne

Abstract

Spreadsheet applications are currently the most prevalent end user tool in organisations across the world. Surveys on spreadsheet use show spreadsheets are used as decision making tools in a range of organisations from credit liability assessment in the business world to patient cardiovascular-anaesthesia risk in the medical community.

However, there is strong evidence to suggest a significant proportion of spreadsheets contain errors that affect the validity of their operation and results. In addition most end users receive no relevant information systems training and consequently have no concept of creating reliable software. This can result in poorly designed untested spreadsheets that are potentially full of errors.

This thesis presents an alternative novel modelling technique to decision support spreadsheets. The novel technique uses attribute classifications (user defined examples) to create a model of a problem. This technique is coined "Example Driven Modelling" (EDM).

Through experimentation, the relative benefits and useful limits of EDM are explored and established. The practical application of EDM to real world spreadsheets demonstrates how EDM outperforms equivalent spreadsheet models in a medical decision making spreadsheet used to determine the anaesthesia risk of a patient undergoing cardiovascular surgery.

This thesis is dedicated to the memory of

Sylvia Joy Thorne

1945 to 2007

You told me I could do it

Praised be our lord for the turn of the year,
For new-born life up-springing;
For buds and blossoms, for lambs and babes,
For thrush and blackbird singing.
May praise, like the lark, leap up from our hearts
To heaven's gate up-winging

Spring song from
The little white horse

Acknowledgements

First of all, I would like to thank Dr David Ball for his guidance, mentoring and friendship during the course of my PhD. Without his supervision and enthusiasm for my research, completing my PhD would have been impossible.

I would also like to express deep gratitude to Mr. Pat Cleary for his clarity of thought, willingness to read, encouragement and sound advice. Pat was particularly instrumental in the final versioning of my thesis.

I would also like to thank Mr. David Chadwick of Greenwich University for his role in my supervisory team. Thanks must also go to the European Spreadsheets Risks Interest Group (EuSpRIG) community who have provided me with valuable feedback on various papers over the years.

I would like to thank Mike, Fi, Katy and Roy for their support throughout the entire length of my study, but especially recently - a difficult time for all of us.

I would like to thank Zoe for all the love, support and friendship you have given me and for having the time to chat about my work or anything else that was bothering me.

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree

Signed.....(Simon Roy Thorne – Candidate)
Date.....

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by giving explicit references. A bibliography is appended

Signed.....(Simon Roy Thorne – Candidate)
Date.....

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and inter library loan, and for the title and summary to be made available to outside organisations.

Signed.....(Simon Roy Thorne – Candidate)
Date.....

Table of Contents

Table of Figures, Equations and tables

Figures

Equations

Tables

1.0 Introduction

1.1 Introduction

This section provides the background to the thesis stating the motivation and background to spreadsheet error. The research question, aim and objectives are defined, the broad research approach is defined and an outline of the thesis is provided.

1.2 Motivation

The motivation for this thesis was the realisation that spreadsheet errors are both prevalent and have significant impact. Further, since relatively little research has been conducted in spreadsheet errors, potentially there is greater opportunity for novel research.

In particular combining spreadsheets with some form of machine learning technique was of particular interest. Potentially machine learning techniques could be used to reduce some of the errors found in spreadsheets.

1.3 Background

End User Development (EUD) describes the activity of end users creating end user applications and information systems using end user software. End user software includes but is not limited to word processing software, spreadsheet software, database software and presentation software.

Of these 'office' type applications, the most prolific is spreadsheet software as noted by several authors (Davies 1987, Jenne 1996, Taylor *et al.* 1998 and Panko and Halverson 1997).

The most recent statistics are taken from the Spreadsheet Research Engineering Project (SERP) who recently surveyed end user development use. Figure 1.1 summarises the main findings regarding the use of end user tools. As can be seen spreadsheets are the most prolific with 99.3% of respondents indicating so.

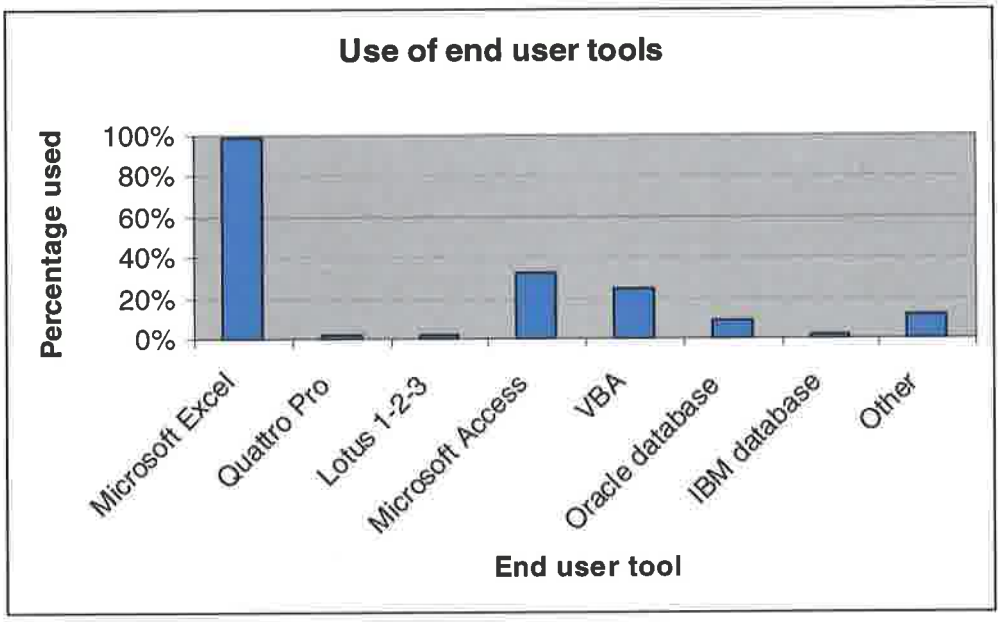

Figure 1.1 End user tool usage (SERP, 2006)

Spreadsheet error rates as summarised by Panko (2006) suggest that the proportion of erroneous spreadsheets is between 30 and 100%.

Considering the prolific use of spreadsheets and the error rates described by Panko, there is opportunity to conduct useful and productive research that may help alleviate some of the problems currently being experienced by the spreadsheet community.

1.4 Research question, aim and objectives

The research question, aim and objectives are outlined below

1.4.1 Research Question

Is it possible to create an alternative modelling technique for the reducion of error in decision support spreadsheets?

1.4.2 Aim

To create and evaluate an alternative modelling technique for the reduction of error in decision support spreadsheets

1.4.3 Objectives

1. Undertake a literature review of relevant topics within the field of spreadsheet error research
2. Based upon the literature review, consider an alternative modelling technique for the reduction of error in decision support spreadsheets
3. Investigate, develop, test and evaluate the proposed novel approach

1.5 Research approach

The purpose of this section is to broadly outline the approach taken towards conducting the primary and secondary research.

The approach taken in this thesis is quantitative and uses the objective scientific method rather than qualitative and subjective immersive study.

Therefore the methods employed in this thesis are influenced by this research philosophy stance, experimentation is used extensively to deductively test the theoretical framework of this thesis.

1.6 Outline of the thesis

Chapter 1 introduces the thesis, outlining the motivation, background, research question, aim and objectives in order to set the scene for the rest of the thesis.

Chapter 2 is a critical review of spreadsheet error, relevant human factors and spreadsheet engineering literature. The literature review emphasises the relationship between human factors and spreadsheet error and uses this relationship to highlight potential areas of novel research.

Chapter 3 introduces the theory of the novel approach and discusses the research design of an experiment which compares the relative advantages of the novel approach with traditional spreadsheet modelling. The results of the experiment are then discussed and tested for statistical significance.

Chapter 4 discusses how the theory of the novel approach, example-giving, could be implemented using an inductive learning approach. The chapter concludes that neural networks are a viable and promising means to implement example-giving. Further, a detailed configuration and design of the type of neural network is considered and standard design for experimentation is suggested.

Chapter 5 establishes some of the important parameters of example-giving when implemented with neural networks. The combination of example-giving and neural networks is termed Example Driven Modelling (EDM). The experimentation in this chapter deals with issues critical to the usefulness of EDM, these include: The number

of examples needed; The effect of complexity on performance; the sensitivity of the learning process (repeatability and consistency) and the effect of noise on performance.

Chapter 6 applies the novel approach, EDM, to 'real-world' spreadsheets to gauge the relative advantages that can be gained from using EDM. In this chapter medical spreadsheets are shown to be a potential area of application for EDM, an example medical spreadsheet is chosen and shown to be erroneous. The medical spreadsheet is then modelled using EDM, the results show that EDM offers significant advantage over the equivalent spreadsheet. Lastly the implications of the successful application of EDM are considered for other professions where similar spreadsheets exist.

Chapter 7 summarises the contributions, conclusions and areas for further work. The main contribution is defined and a summary of the most significant conclusions is presented. Finally some areas for further work are suggested that would further extend the research contained in this thesis.

2.0 Spreadsheet Error

2.1 Overview of the chapter

Section 1.4.3 objective 1 stated:

> *Undertake a literature review of relevant topics within the field of spreadsheet error research*

Section 2.3 examines statistical studies on spreadsheet error and the error rates quoted from those studies. Section 2.4 reviews the error types and taxonomies quoted in spreadsheet error literature. Section 2.5 examines the relationship between human factors and spreadsheet error with reference to quantifiable and unquantifiable human factors. Section 2.6 investigates spreadsheet error reduction techniques. Section 2.7 explores the evidence suggesting that spreadsheets error is influenced by wider human – computer interaction issues. Section 2.8 highlights areas of novel research in light of the literature review and section 2.9 summarises the spreadsheet errors literature review.

2.2 Introduction to spreadsheet Error

A spreadsheet error could be an unintentional mistake or omission which causes part or all of a spreadsheet, or spreadsheet model, to become erroneous. The term

'spreadsheet model' and 'spreadsheet' are used interchangeably in this thesis and can defined as models of real world problems, business or otherwise, created using spreadsheet software.

Spreadsheet error is evident in at least 30% of all spreadsheet models (Panko, 1999). An example of the impact a spreadsheet error can have in industry is the loss of $24 Million by Trans Atlanta Corporation due to a copy and paste error when using a spreadsheet to bid for energy contracts in New York, USA (EuSpRIG, 2006). The loss experienced by the Trans Atlanta Corporation is one example of many where spreadsheet errors have caused significant financial loss in organisations.

2.3 Error statistics

Academic interest in spreadsheet error has increased, judging by the increase in academic papers published in journals and conferences concerning spreadsheet errors.

Spreadsheet error research has yielded statistical and case based studies on spreadsheet error. Statistics produced on spreadsheet errors report varying error rates and use different metrics.

The most commonly used metric is "Percentage of models with error" (Panko, 1998) which simply provides a percentage figure which describes the number of spreadsheet models with at least one error.

Another commonly used measure is "Cell Error Rate" (Panko, 1999). Cell Error rate only considers non text cells, i.e. those cells that contain formulae. The error rate is calculated by dividing the number of erroneous formula cells by the number correct formula cells.

Statistics are produced from either lab based experiments or auditing case studies using 'live' spreadsheets gathered from organisations. Lab based experiments are used to explore a particular theory or to prove a particular point whereas audit studies demonstrate error in practice.

Frequently in lab experiments, where there are multiple participants, the term 'percentage of models with error' is used as an overall measure of success or failure. This is calculated by taking the total number of spreadsheet models produced in the experiment and dividing by the number of erroneous spreadsheet models.

The first documented study of spreadsheet error was conducted by Brown and Gould for IBM in 1987 (Brown and Gould, 1987). This study took 9 experienced spreadsheet developers and examined their performance when asked to create a number of spreadsheets from scratch. They found that all participants made at least one error and in total 63% of the models produced were incorrect, as noted by Panko (1999).

Since this original paper, there have been many studies of spreadsheet error yielding varying error rates. Table 2.1 depicts some experimental studies with relevant error rates.

Author(s)	Year	Percentage of models with errors
Hicks and Panko	1995	91%
Javrin and Morrison	1996	84%
Panko and Halverson	1997	80%
Panko and Halverson	1998	86%
Javrin and Morrison	2000	95%

Table 2.1 Error rates in experimental studies (adapted from Panko, 2006)

Error rates contained in table 2.1 show that, in these experimental studies, nearly all spreadsheets contain error. Field audit studies of 'live' spreadsheets record similar error rates. Live spreadsheets are defined as spreadsheets that are in use by an organisation or professional. Table 2.2 contains error rates found in live spreadsheets

Author(s)	Year	Percentage of models with error
Hicks	1995	100%
Coopers & Lybrand	1997	91%
KPMG	1997	91%
Lukasic	1998	100%
Butler	2000	86%
Clermont, Hanin, & Mittermeier	2002	100%

Table 2.2 Error rates in field audits (adapted from Panko, 2006)

Studies such as those contained in table 2.1 and 2.2 have increased awareness of spreadsheet error in some sections of the academic and business community. Naturally these studies on spreadsheet error have led researchers to attempt to classify error types observed in the studies.

Defining a taxonomy of spreadsheet error has been attempted by several authors (Panko and Halverson 1998, Rajalingham *et al.* 2000, Rajalingham 2005, Purser and Chadwick 2006). However, there is no consensus between the authors although they all loosely base their work on Panko and Halverson's 1998 paper.

2.4 Spreadsheet errors types

Panko and Halverson (1998) split spreadsheet error into quantitative and qualitative types. Within the quantitative error type, Panko and Halverson (1998) discuss three areas of 'known error': Mechanical, Logical and Omission. No detailed explanation is given of the qualitative error type.

Panko and Halverson's definitions of error types are heavily influenced by human error taxonomies such as Reason (1990) and Allwood (1984).

2.4.1 Mechanical error

Mechanical error in spreadsheets, according to Panko and Halverson (1998)

> *"Mechanical errors are simple mistakes, such as mistyping a number or pointing to the wrong cell"*

From this definition, one can conclude that both mistyping and incorrect cell referencing are mechanical error. However, it is not clear if syntactical errors are mechanical errors or logic errors, i.e. mistyping the syntax of a command.

2.4.2 Logic error

Panko and Halverson (1998) define logic error as:

> *"Logic errors involve entering the wrong formula because of a mistake in reasoning"*

Logic error is therefore based upon the domain knowledge of an individual and the implementation of that knowledge in a spreadsheet. Panko (2005) notes that logic errors are harder to detect and correct than mechanical errors.

2.4.3 Omission error

Panko and Halverson (1998) define omission error as *"when something is left out"* and comment that this type of error is the most *dangerous* and is the most difficult to detect, a point of view shared by Colver (2007).

Given the definition, omission error can account for anything that is left out of the spreadsheet. This may be the omission of a cell containing a figure in a sum calculation or it may be the omission of a constraint in a rule. This means that omission error has a very broad definition and potentially crosses over with other error types.

2.4.4 Taxonomies of spreadsheet error

Panko and Halverson's (1998) definitions of error types give broad guidelines to define error types but lack detailed definitions. The authors present the work as an observation on common errors rather than an exhaustive taxonomy of spreadsheet errors.

Several researchers have built upon Panko and Halverson's (1998) error types to create taxonomies of spreadsheet error (Rajalingham *et al.* 2000, Rajalingham 2005, Purser and Chadwick 2006).

Rajalingham *et al.* (2000), see figure 2.1, provides a taxonomy of spreadsheet errors directly influenced by the error types in Panko and Halverson (1998). This detailed taxonomy further defines mechanical, logic and domain errors in a decision tree structure.

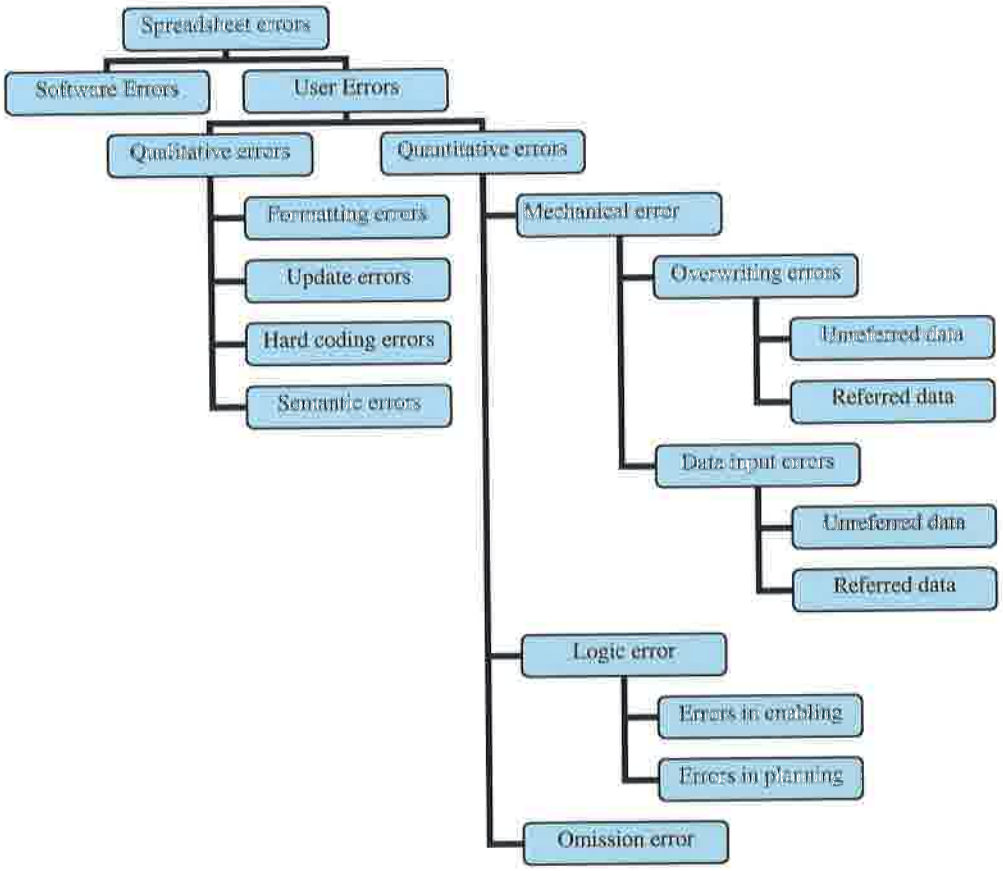

Figure 2.1 A taxonomy of spreadsheet error Rajalingham (2000)

Rajalingham (2005) extends Rajalingham *et al.* (2000) with a revised classification of spreadsheet error, see figure 2.1. The rationale for revising Rajalingham *et al,* (2000) was:

> *"...the derivation and the justification (of the last taxonomy) was not discussed in adequate detail"*

However, Rajalingham (2005) discards the widely cited three error types, Mechanical Logic and Omission (Panko and Halverson 1998), and uses new terms such as *Accidental and Reasoning* and detailed specifications of errors.

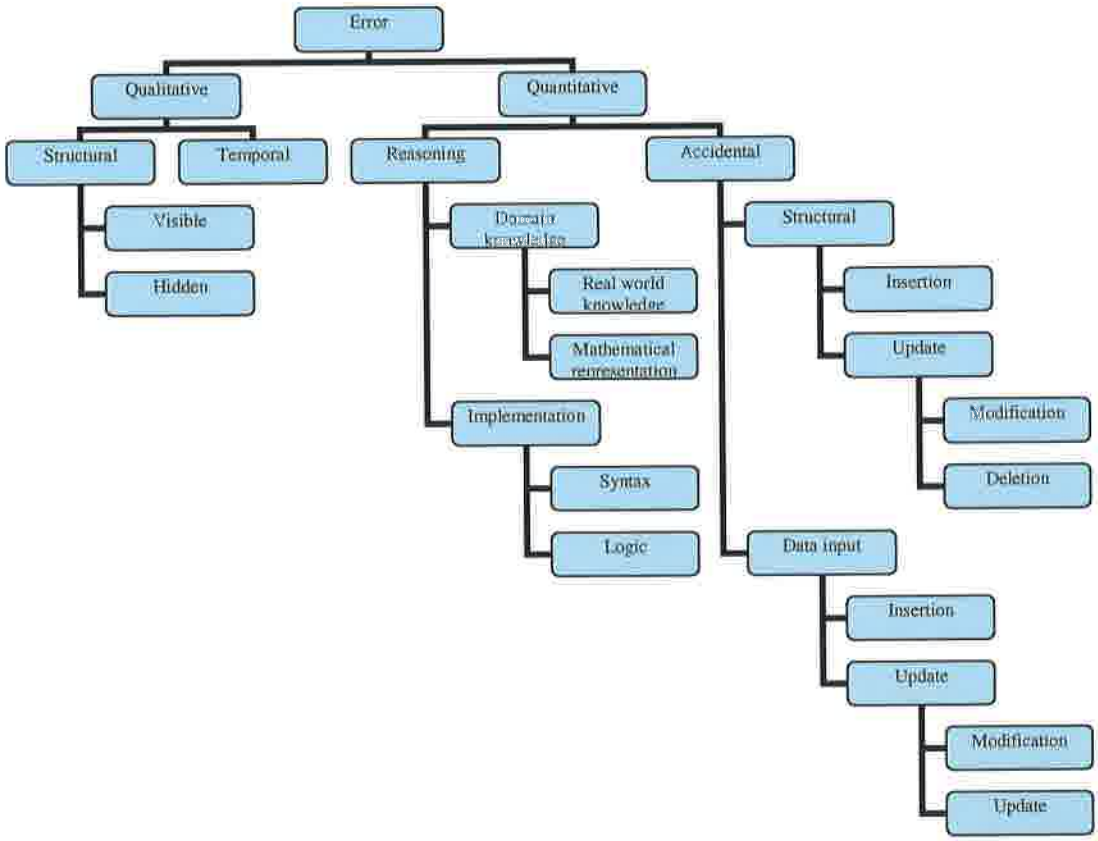

Figure 2.2 A revised taxonomy of spreadsheet error Rajalingham (2005)

This detailed specification answers some of questions arising from error types defined in Panko and Halverson (1998). However, this taxonomy is over prescribed and conflicts in error type identification arise in some rules. In particular, a crossover between 'structural' and 'data entry' errors exists, as Purser and Chadwick (2006) discuss:

> *"It could be argued that a potential data input error is actually caused by a structural error when the developer fails to create a robust structure (formula network) in the spreadsheet"*

2.4.5 Conclusions on spreadsheet error literature

Examples of current spreadsheet error research comprises experiments, taxonomies of spreadsheet error, observations of spreadsheet error in practice, theories on spreadsheet error management, manual auditing and auditing software.

Experiments concerning spreadsheet error offer some quantification of the magnitude of spreadsheet error and impact that errors can have on organisations. Although error rates vary, it is clear from the evidence available that at least 30% of spreadsheets contain error.

Taxonomies of error (Rajalingham 2000 and Rajalingham 2005) provide a means to classify error. However, inconsistency in terms (Rajalingham, 2005), discrepancies in error classification (Rajalingham 2005) and prescriptive structures (Rajalingham *et al.* 2000, Rajalingham 2005) make these taxonomies problematic to apply. Judging by the number of citations to Panko and Halverson's (1998) original paper, Panko and Halverson's error types are apparently easier to apply.

Similarities exist between spreadsheet error types and taxonomies (Panko and Halverson 1998, Teo and Tan 1999, Ayalew *et al.* 2000, Rajalingham *et al.* 2000 and Rajalingham 2005) and human error taxonomies (Rasmussen 1974, Allwood 1984, Reason 1990 and Norman 1980).

Further, Panko (2006) argues that spreadsheet errors are complex and similar to human error. It has been suggested by Panko (2006) that human and spreadsheet errors are closely related. Moreover, Panko suggests that spreadsheet error rates are approximately the same as error rates found in other human activities such as spelling, typing or programming a computer.

Therefore considering the relevant human factor literature is important in understanding and mitigating spreadsheet errors. Consequently we consider the relevant human factors.

2.5 Relevant human factors

Human factors are synonymous with ergonomics and human error. Human Factors as defined by Ergoweb (2007):

> *"A term synonymous with 'ergonomics', [human factors] is the branch of this science that began in the US and focuses on cognitive performance of humans"*

Human factor and specifically human error research is established in safety critical literature such as Nuclear Power (Swain and Guttman, 1983) and Aviation (Wiener and Nagel, 1988).

Further, it is well recognised that human factors have a relationship with error and misjudgements of professionals (Reason 1990, Reason 2005, Rasmussen 1974, Allwood 1984 and Norman 1980).

The impact on quality arising from human factors has emerged in non safety critical literature such as software engineering. One such example is the numerous detailed studies on errors per line of code (Akiyama 1971, Basili and Selby 1986, Bush 1990 Jones 1998).

Panko adapts themes from the study of human factors in software engineering and applies them to spreadsheets. (Panko and Halverson 1998, Panko 1999, Panko 2003, Panko 2007). Exploring human factors in spreadsheet development is also considered in Thorne and Ball (2005a) and human factors in end user development in Thorne and Ball (2005b)

Within spreadsheet literature, human factors divide into two distinct areas: Quantifiable human factors and Unquantifiable human factors. The next section discusses those topics already examined in spreadsheet literature and considers other relevant human factor research.

2.5.1 Quantifiable human factors

Quantifiable human factors are defined as measurable effects arising from human factors in spreadsheets. Examples include: Base Error Rate; Cognitive Load; Miller's threshold as discussed below.

2.5.1.1 Base Error Rate

Base Error Rate (BER), which is also referred to as Basic Error Rate, is described by Panko (1999) as:

> *"A background error rate committed by humans in simple and repetitive tasks"*

Evidence gathered by Panko (2005) presents BERs which vary depending on the task. For example a typical BER observed in spelling ranges from 0.5% to 2.4% (errors per word). See table 2.3 for some examples of error rates in simple tasks.

Study	Detail	Error Rate
Chedru & Geschwind (1972)	Grammatical errors per word	1.1%
Baddeley & Longman (1978)	Entering mail codes. Errors after correction. Per mail code.	0.5%
Grudin (1983)	Error rate per keystroke for six expert typists. Told not to correct errors, although some did. Per keystroke.	1%
Hotopf (1980)	W sample (written exam). Per word	0.9%
Wing & Baddeley (1980)	Grammatical errors in examination at Cambridge. Per word.	0.5
Mitton (1987)	Study of 170,016 errors in high-school essays, spelling errors. Per word.	2.4%

Table 2.3 BER in simple tasks (Adapted from Panko 2005)

Programming tasks yield a higher BER, evidence collected from studies of programming show a BER between 2% and 9%, see table 2.4

Study	Detail	Error rate
Akiyama [1971]	17,052 lines of assembler code. 6 modules average 2,842 lines.	2.0%
Basili & Selby [1986]	20 KLOC FORTRAN. Mixture of new and old code.	1.1%
Fagan [1976]	Aetna Life and Casualty, 4,439 lines of non-comment code. Found during code inspection.	3.8%
Graden & Horsley [1986]	Major telecommunications project at ATT. 2.5 million LOC over 8 software releases.	3.7%
Jones [1986]	Five systems at AT&T	5.0% - 9.5%

Table 2.4 BER in complex tasks (adapted from Panko 2005)

Several authors have suggested that producing spreadsheet models is akin to programming a computer (Rajalingham *et al.* 2000, Panko 2005). On that basis spreadsheet modellers more likely to commit a 5% BER in spreadsheet tasks, as Panko (2005) notes.

Comparing table 2.3 and 2.4, it appears that the more complex the task, i.e. programming is more complex than data input, the higher the associated BER.

However, research conducted by Takaki (2005) suggests that the relationship between complexity and BER is also affected by the modellers self efficacy. Self efficacy is defined by Bandura (1994) as:

> *"Perceived self-efficacy is defined as people's beliefs about their capabilities to produce designated levels of performance that exercise influence over events that affect their lives...such beliefs produce these diverse effects through four major processes. They include cognitive, motivational, affective and selection processes."*

Takaki (2005) found that BER was affected in a complex manner by the self efficacy of each modeller.

2.5.1.2 Miller's threshold

Miller (1956) considered human working memory and its limitations in his seminal work "the magical number seven plus or minus two". Miller demonstrates that unaided humans start to make errors in calculation when they are dealing with seven plus or minus two concepts simultaneously.

Considering the problematic syntax and the abstract nature of programming formulae in spreadsheets (Napier *et al.*, 1989), Miller's threshold is important.

Whilst there are no explicit guidelines on spreadsheets for Miller's threshold, one could view "concepts" on a cell-by-cell basis. Using that system, "concepts" would be elements of a formula in a cell.

Considering the complexity of spreadsheet applications (Napier *et al.* 1989), spreadsheet formulae may well routinely breach Miller's threshold (Thorne *et al.* 2004).

This is a grey area due to a lack of research although *working memory* as discussed by Miller (1956) is incorporated in cognitive load theory.

2.5.1.3 Cognitive load

Sweller (1994) defined cognitive load as the amount of "cognitive pressure" exerted on a human being when undertaking an activity. Sweller (1994) states that the higher the cognitive load, the more difficult the task, the greater the likelihood of error.

According to Sweller (1994) and Kruck *et al.* (2003) cognitive load theory is based upon four interlocking supersets: Skill Character; Working memory; Long-term memory and Task Demand. These supersets contain subsets, such as problem solving, memory load and accuracy. Assessing each subset in each superset allows one to calculate the cognitive load of a task.

Kruck *et al.* (2003) applied this method to a number of different tasks that ranged from typing to routine medical diagnostics. The authors also applied this method to spreadsheet tasks, the results indicate a high cognitive demand for spreadsheet tasks, see table 2.5.

	Skill Character			Working memory		Long term memory		Task demands	
Tasks	Problem solving	Perceptual motor	Planning	Unit task structure	Memory load	Input to LTM	Retrieval from LTM	Pacing	Accuracy
Typing	Low	High	Low	Low	Low	Low	Low	Low	Int.
Driving a car	Low	High	Int.	Low	Int.	Low	Low	High	High
Mental multiplication	Int.	Low	Low	High	High	Low	Int.	Low	High
Balancing check book	High	Low	Int.	High	High	Int.	Int.	Low	High
Writing a business letter	High	Low	High	High	Int.	Int.	Int.	Low	Int.
CPA doing income tax	High	Low	High	High	Int.	Int.	High	Low	High
Routine medical diagnostics	High	Low	High	High	High	High	High	Int.	High
Spreadsheet tasks	**High**	**High**	**High**	**High**	**High**	**Low**	**High**	**Low**	**High**

Table 2.5 Cognitive load analysis (Kruck et al., 2003)

According to Kruck *et al.* (2003) spreadsheet tasks and routine medical diagnosis have a similar cognitive load. This high cognitive demand for spreadsheet tasks means that spreadsheet modellers are more likely to commit errors (Sweller, 1994).

Further, consider that routine medical diagnosis exerts a high cognitive demand but individuals performing routine medical diagnosis are extensively trained professionals.

In contrast, most spreadsheet modellers receive little or no formal training (Davies 1987, Alavi and Weiss 1985, Munro *et al* 1987, Alavi *et al* 1987, Taylor *et al,* 1998, SERP 2005 and Pemberton and Robson 2000). This observation may explain the compounding of some errors.

On that theme, Kruck *et al.* (2003) studied the effect of training a spreadsheet modeller with skills that could help them deal with the cognitive load exerted in spreadsheets.

Kruck *et al.* (2003) found that the only element that improved significantly after training was logical deduction. Kruck *et al.* found that by improving the participant's logical reasoning skills, the modellers committed fewer errors.

2.5.2 Unquantifiable human factors

The unquantifiable human factors are less tangible and more difficult to measure than the quantifiable human factors.

The unquantifiable human factors affect perception of quality and reliability in individuals and groups. This may result in overconfidence or bias.

As shown below, the unquantifiable human factors divide into two topics, the first deals with overconfidence in spreadsheet modellers and the second deals with bias and its affect on spreadsheet modellers.

2.5.2.1 Overconfidence

Overconfidence is defined by The Oxford English Dictionary (2006)

> *"Excessive confidence; greater confidence than is warranted."*

Overconfidence is prolific in human activities as Koriat *et al.* (1980) demonstrates with problem solving:

> *"Problem solvers and planners are likely to be overconfident in evaluating the correctness of their knowledge"*

Russo and Schoemaker (1989) consider the costs and causes of overconfidence in decision making. Russo and Schoemaker (1989) administered a quiz across a range of industries to investigate how overconfident individuals were. The research showed that all individuals across all industries were overconfident, rates of overconfidence ranged between 42 and 80%.

Overconfidence does not only apply to novice or inexperienced professionals. The same rules apply to 'experts' as Lusted (1977) and Oskamp (1965) demonstrated with physicians and clinical psychologists respectively.

Research into spreadsheet modellers and overconfidence has shown that both individuals and groups demonstrate chronic overconfidence (Panko, 2003). Panko found that 80-100% of the modellers he examined were overconfident in the quality of work produced.

Other evidence of overconfidence in spreadsheet development includes Brown and Gould (1987) who examined confidence in models produced by nine experienced spreadsheet modellers. All nine modellers indicated they were "very confident" despite 63% of the models being erroneous.

Further, Davies and Ikin (1987) and Floyd and Pyun, (1987) note that when spreadsheet modellers are asked to indicate the quality of their work, most answered they were "confident".

From the available evidence it is apparent that spreadsheet modellers, novice or experienced, suffer from overconfidence when developing spreadsheet models. The effect of this overconfidence could result in a lack of planning, testing and future maintenance for a spreadsheet.

2.5.2.2 Bias

The oxford English dictionary (2007) defines bias as:

"To give a bias or one-sided tendency or direction to; to incline to one side; to influence, affect (often unduly or unfairly)"

There are several biases that may affect spreadsheet modellers significantly although no research has been carried out to identify particular biases in spreadsheet modelling.

2.5.2.2.1 Optimism bias

Gilovitch *et al.* (2002) and Armor and Taylor (2000) define optimism bias as:

"...the tendency of predicting a conclusion favourably where the subject has a vested interest in the outcome"

Gilovitch *et al.* (2002) consider optimism bias in their work on heuristics and causality in decision making. Armor and Taylor (2000) note optimism bias in their work on self regulation and perception.

In spreadsheet modelling, optimism bias may cause an inaccurate perception of what can be achieved by an individual or a group with the tools and resources available.

Modellers with optimism bias may incorrectly use spreadsheets, where in reality the task is too complex, too critical or resources too little.

Whilst there has been no direct identification of this optimism bias in spreadsheet literature, one possible example of such a case is taken from Butler and Croll (2006) who investigated the use of spreadsheets in clinical medicine.

Butler and Croll (2006) found one spreadsheet in particular for calculating dosage levels used in anaesthesia based upon several clinical input measures.

Whilst no material errors were found, using an error prone spreadsheet for anaesthesia dosage calculations is risky and demonstrates optimism bias, i.e. the belief that a spreadsheet is robust enough to be used for safety critical calculations.

2.5.2.2.2 Hypothesis Fixation

Fraser and Smith (1992) discus hypothesis fixation in human activities using the definition provided by Wason (1960).

> *"Hypothesis fixation occurs when a subject maintains a hypothesis that has been demonstrated to be false"*

There is no acknowledgement in spreadsheet literature of hypothesis fixation. However, if one considers how spreadsheets are used, the possibilities of hypothesis fixation causing an error in judgement are substantial.

Spreadsheets are developed relatively quickly and are considered a "scratch pad" application (Panko 2005, Grossman 2002). i.e. spreadsheet modellers can use spreadsheets to model ideas quickly and get an impression on the likely results.

The potential for hypothesis fixation to arise based upon some initial analysis using a scratch pad spreadsheet may be significant.

2.5.2.2.3 Confirmation Bias

Confirmation bias is defined by Fraser and Smith (1992) as:

> *"Confirmation bias arises when individuals test their own work with test conditions and data that favour a positive response"*

In other words confirmation bias causes the modeller to test a model with data sympathetic to the design, i.e. it will not objectively test the validity of the model.

So the user may "test" the spreadsheet but only with data that shows the spreadsheet to be working correctly. In this vein, confirmation bias may have a relationship with overconfidence.

2.5.2.3 State space searching

Newell and Simon (1972) suggest problem solving in humans can be viewed as problem state space searching.

Problem state space searching is the process of forming a goal state (what the user wants to create) a current state (the point that the user currently resides at) and the valid operators to change the current state to the goal state.

The goal state in spreadsheet modelling could be generic or specific; it could be to create a spreadsheet that represents a business problem or the sum of two cells.

Consider the latter example, the goal state is a formula that sums two cells; the current state is nil (there is no part of the formula produced).

The valid operators are mathematical symbols (+ - / *), cell names and addresses (C1, B1 etc) and the applications specific operators (SUM).

In this instance the problem space allows more than one valid goal state, there are several ways of writing a formula that will sum two cells. It is now at the modeller's discretion to decide on which goal state to employ.

Selecting the best goal state presents the user with some significant problems. How does the user decide which is the *best* solution to the problem or are they even aware that there are other valid goal states. Research shows that spreadsheet modellers know few of the commands available in spreadsheet software (Napier *et al.*, 1992).

2.5.3 Further discussion on spreadsheet error

As shown in section 2.2, spreadsheet literature shows us that spreadsheet errors exist and can cause material loss.

Some authors view spreadsheet error as an organisational problem, i.e. the way forward is to control use via policy (Madahar *et al.,* 2007) and best practice in the organisation (Grossman, 2002). Others view spreadsheet errors as a technical problem, i.e. develop new technology that reduces the risk of spreadsheet errors (Clermont and Mittermeir 2002, Paine 2007)

Section 2.5 suggests that human errors have a close relationship with spreadsheet errors. Speculatively, a more all encompassing perspective suggests that factors (human, organisational and technical) may even overlap as suggested by figure 2.3

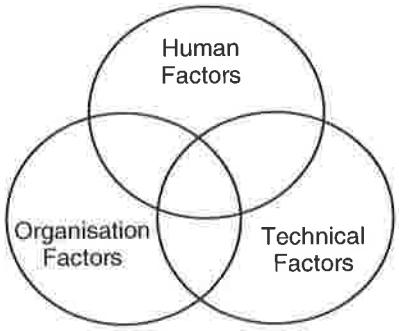

Figure 2.3 Sources of spreadsheet error

The majority of spreadsheet literature is written from an organisational or technical perspective. This has led to research that attempts to reduce spreadsheet errors either by organisational or technical means.

There has been little research into managing spreadsheet errors either from an organisational or technical viewpoint that identifies the need to manage human factors, or considering all three factors. This would be a good area for further research.

2.5.4 Conclusions on relevant human factors

Human factors play a significant role in spreadsheet errors and have been largely ignored by the wider spreadsheet community. Some authors (Panko, 2007) suggest that spreadsheet errors are human errors found in spreadsheets.

The quantifiable human factors such as BER (Panko, 2005) and Cognitive load (Kruck *et al.* 2003) have been shown to have an effect on spreadsheet quality. Other aspects such as Miller's threshold (Thorne *et al.* 2004) bear on the relationship between spreadsheets and error

The unquantifiable human factors overconfidence and bias significantly effect spreadsheet model quality (Burnett *et al.* 2003, Panko 2003, Brown and Gould 1987, Davies and Ikin 1987, Floyd and Pyun 1987).

State space searching (Newell and Simon (1972) may also affect spreadsheet error considering the extent of spreadsheet modellers knowledge of spreadsheet syntax (Napier *et al.*, 1992).

Overconfidence has also proven difficult to mitigate (Burnett *et al.* 2003, Panko 2003).

2.6 Error reduction methods

Research on error reduction in spreadsheets is the other distinct subject that exists in spreadsheet error research.

As discussed below, error reduction research includes lab based auditing, modified software engineering methodologies and automated tools.

Studies of error reduction, usually emphasise the effectiveness of an approach, i.e. the effectiveness of a method at preventing, detecting or reducing error in spreadsheets.

This includes subjects as diverse as: Manual auditing; Software engineering principles and error reduction or prevention software.

2.6.1 Manual auditing methods

Manual auditing is the process of manually checking spreadsheets after creation using the spreadsheet application and the skill of the auditor. Manual auditing has two separate approaches: individual audit and team audit.

2.6.1.1 Individual auditing

Individual auditing is the process of an individual auditor checking a spreadsheet for mistakes.

Research into individual spreadsheet audits follows a similar research methodology. Typically studies present participants with a spreadsheet seeded with errors which they are required to audit. The researcher then measures the effectiveness of the audit based on the number of mistakes detected and corrected.

Galletta *et al.* (1993) presented the participants of the study with a model seeded with errors. The participants were asked to analyse the model and find the errors. The study sampled a range of spreadsheet experience.

Galletta *et al.* (1993) found on average that 56% of the seeded errors were discovered. Interestingly the study found that experienced spreadsheet users were quicker at auditing the model but not significantly more accurate.

Galletta *et al.* (1997) extended Galletta *et al.* (1993) with a larger scale study, the same task was used as the 1993 experiment and a similar detection rate of 51% was observed.

Panko (1999) conducted a similar auditing experiment to that of Galletta (1993), Panko found that individual auditors found 63% of seeded errors.

Howe and Simkin (2006) offer a similar experiment to that of Galletta (1993) and Panko (1999). In this study participants were asked to audit an error seeded spreadsheet. On average 67% of errors were detected and statistical analysis of demographic information revealed that age, academic ability and level of education all improved the participant's ability to detect errors.

From the available literature (Galletta *et al.* 1993, Galletta *et al.* 1997, Panko 1999, Howe and Simkin 2006) error detection rates using auditing detect approximately 50-60% of seeded errors. See table 2.6 for a summary of individual audit experiments.

2.6.1.2 Team auditing

The team or peer audit differs from individual audits since several auditors work on the same spreadsheet. Researchers, such as Panko (1999), suggested that team auditing may have a better error detection rate than individuals.

Study	Subjects	Sample	% errors detected	Notes
Galletta *et al.* (1993)	MBA students & CPAs Taking a Continuing Education Course	60	56%	Budgeting task containing seeded error. (worked alone)
Galletta *et al.* (1997)	MBA students	113	51%	Same task used 1993 study
Panko (1999)	Undergrads			Modified version of Galletta wall task.
	Undergrads working alone	60	63%	Working alone
	Undergrads working in groups of three	60	83%	Team working
Howe and Simkin (2006)	Undergrads	228	67%	Worked alone

Table 2.6 Audit experiments summary (Adapted from Panko 2005)

Panko (1999) found that team auditing (groups of three) found 83% of errors as opposed to 63% with individual auditors, a finding echoed by Vemula *et al.* (2006). Based on this evidence, team auditing appears to be more effective than individual auditing.

The principle of team auditing being superior to individual auditing was recently demonstrated at the EuSpRIG 2007 annual conference. A simple error seeded spreadsheet was given to nine individuals and one group of three to audit. After an hour of auditing individuals on average found 43% of errors and the group of three found 74% of errors.

2.6.2 Software Engineering methods

Researchers have sought to apply software engineering methods to spreadsheets to reduce error (Rajalingham *et al.* 2000, Burnett *et al.* 2001, Grossman 2002, Burnett *et al.* 2003, Yirsaw *et al.* 2003, Burnett *et al.* 2004, Grossman and Ozluk 2004, Prior 2004, Panko 2006).

Considering that the origins of software engineering were the software crisis, i.e. poor quality software, adapting such techniques to spreadsheets is a promising idea.

The focus of software engineering research in spreadsheets varies, some researchers investigate spreadsheet best practice (Grossman, 2002), some investigate structured development in spreadsheets (Burnett *et al.*, 2003) and others examine spreadsheet testing (Panko, 2006).

Some researchers (Grossman 2002, Burnett *et al.* 2004) suggest that spreadsheet error can be managed by adapting software engineering methods to create a special discipline called "spreadsheet engineering".

2.6.2.1 Spreadsheet Engineering

Grossman (2002) presents eight principles for a spreadsheet engineering methodology, see figure 2.4.

1. Best practice can have a large impact
2. Lifecycle planning is important
3. A priori requirements specification is beneficial
4. Predicting future use is important
5. Design matters
6. Best practice is situation dependant
7. Programming is a social, not an individual activity
8. Deployment of best practices is difficult and consumes resources

Figure 2.4 Spreadsheet engineering principles (Grossman 2002)

Grossman discusses challenges in a spreadsheet engineering methodology, identifying the need for a taxonomy of spreadsheet user experience. Grossman (2002) notes that a consensus on practices to improve spreadsheet quality is difficult, since quality in spreadsheet modelling is a subjective issue. A notion supported by Colver (2004) who remarks that "*best practice*" is a contentious issue since the application of *best practice* approaches often contradict one another.

Rajalingham *et al.* (2000) discuss a structured methodology for spreadsheet modelling using data diagrams and modularisation of spreadsheets. Rajalingham *et al.* (2000) consider traditional approaches to data modelling through the use of Entity Relationship (ER) diagrams (Chen, 1975) and Jackson's Structured Diagrams (JSD) (Jackson, 1975).

The research demonstrated that using JSD, spreadsheets can be modularised which aids the future maintenance of the model. However, this approach would require spreadsheet modellers to know how to apply JSD and in practice would require training.

Burnett *et al.* (2003) discusses end user software engineering in spreadsheets giving "user assertions" (pointers to potential errors) for debugging a spreadsheet. The assertions assisted the spreadsheet modellers to detect and correct spreadsheet errors that would have otherwise been missed.

Burnett *et al.* (2004) suggest that since spreadsheet modellers are not IS professionals, it is more practical to use a small feedback loop rather than a comprehensive SDLC

based approach. The results of their experiment showed that spreadsheet modellers found this new feedback approach easier to put into operation than a strict software engineering approach.

Grossman and Ozluk (2004) extend previous work on spreadsheet engineering principles, Grossman (2002), to give a more traditional adaptation of the SDLC, see figure 2.5

1. Modelling
2. Development parameters
3. Design
4. Programming
5. Quality Control
6. Debugging
7. Documentation
8. Usage
9. Modification

Figure 2.5 Revised spreadsheet engineering principles (Grossman and Ozluk, 2004)

This spreadsheet engineering framework takes special consideration of how spreadsheet models are used in practice. In particular stages 8 (usage) and 9 (modification) acknowledge that the use of a spreadsheet may change and that the spreadsheet may be modified in the future.

However, Grossman and Ozluk (2004) provide only the theoretical benefits and do not include any data to indicate if this approach improves quality in practice.

Rust *et al.* (2006) demonstrate the potential of Test Driven Development (TDD) as a means to develop spreadsheets. Rust *et al.* (2006) found that by using TDD to develop spreadsheet models, potentially the quality of the resulting spreadsheet model increased.

In conclusion, spreadsheet engineering is grounded in software engineering principles that can provide theoretical benefits, however establishing 'spreadsheet engineering' will take more time and research.

2.6.2.2 Spreadsheet Testing

There are few studies which provide testing strategies for spreadsheets, although testing is identified as an important consideration for reducing error (Panko, 2006).

Burnett *et al.* (2001) provides a testing methodology designed to help users locate errors before the model is implemented. Burnett *et al.* (2001) combines a testing methodology with an interactive audit tool which indicates potentially erroneous cells on the spreadsheet.

Pryor (2004) adapts software engineering tests (Pressman and Ince, 2000) to synthesize a technique suitable for spreadsheets.

Pryor suggests the following tests: Unit testing (individual units as the spreadsheet is developed); System testing (the performance of the spreadsheet as a whole); Regression testing (comparing the spreadsheet with its predecessors) and Acceptance testing (user acceptance of the spreadsheet).

Whilst this method is proven in software engineering (Pressman and Ince 2000), Pryor (2004) offers no data to suggest that such an approach is effective in spreadsheets.

Yirsaw *et al.* (2003) describe a software engineering debugging method applied to spreadsheets. Yirsaw *et al.* (2003) apply a method called 'interval based testing' to spreadsheets. Interval based testing is a 'dynamic' testing method intended to be used as the spreadsheet is being developed.

Yirsaw *et al.* (2003) argue that interval based testing allows spreadsheet modellers to detect errors before they are implemented. However, no practical evidence is provided to show that modellers can use this approach or any indication of spreadsheet error detection rate.

Panko (2006) summarises testing techniques for spreadsheets and recommends a strategy for spreadsheet testing. This strategy advocates that testing should account

for 25-40% of all spreadsheet development time. It also suggests that specific testing methodologies, such as the Fagan method, should be used on planning documents as well as spreadsheets.

The Fagan method (Fagan, 1986) is an iterative testing cycle conducted by a team of testers. The Fagan method has been shown to reduce defects by 80 to 90 percent (Fagan, 1986).

2.6.3 Software tools

Research into spreadsheet auditing and testing has led to development of partially automated software tools. These tools appear as software add-ins to the standard spreadsheet application. These tools offer automated auditing functions, spreadsheet control functions and alternative spreadsheet programming environments.

2.6.3.1 Spreadsheet auditing software

Spreadsheet auditing software is defined as a third party vendor add in, which performs auditing functions.

Spreadsheet auditing software has two basic methods which we refer to as: standard and specialised. Standard auditing functions include cell dependency tracing, collating and displaying of formulae and indication of potentially erroneous formulae. Typically the auditing software provides a number of graphical representations of the spreadsheet, such as a "Formula map".

Specialised auditing functions have applications in particular industries, usually in addition to standard functions. For example, Spreadsheet Auditing from Customs and Excise (SpACE) as discussed by Butler (2000) performs a variety of "standard" auditing functions and has "specialised" functions that relate VAT calculations.

There are many examples of spreadsheet auditing software, which offer approximately the same specification. However, there are some pieces of software that offer novel features to reduce spreadsheet error.

A novel approach is taken by XLAnalyst, this software offers standard audit functionality but also attempts to quantify spreadsheet risk. The risk calculation is based upon the potential errors found in that spreadsheet.

Research shows that when faced with a large volume of spreadsheets, it is infeasible to audit all of them (Nash and Goldberg 2005, Butler 2000 and Pryor 2004 and 2003).

Software such as XLAnalyst, which could prioritise the spreadsheets according to risk, would address auditing issues raised by many authors (Nash and Goldberg 2005, Butler 2000, Pryor 2004 and Pryor 2003).

However, measuring risk in spreadsheets is particularly difficult (Madahar *et al.* 2007) to date there is no agreed mechanism for measuring risk in spreadsheets.

Research on spreadsheet risk management and classification conducted by Madahar *et al.* (2007) shows some potential on reaching an acceptable risk classification model. However, this model has yet to be evaluated fully in practice.

Clermont presents a tool for auditing spreadsheets in a series of papers (Clermont *et al.* 2002, Clermont 2003, Clermont and Mittermeir 2003 and Clermont 2004).

Clermont's software tool kit allows the user to visualise large spreadsheets. The visualisation process converts spreadsheets into hierarchical and graph based representations. The visualisation is based upon the logical areas of a spreadsheet, the semantic classes and the data modules. Visualisation allows modellers to spot inconsistencies in data and erroneous values.

Clermont *et al.* (2002) demonstrate the visualisation tool kit on spreadsheets gathered from industry. The tool kit found 241 material errors in 3 real-world spreadsheets from a company claiming the spreadsheets were "error-free". However, the actual

auditing was done by the creators of the tool kit, therefore the practical usability of these tools remains unclear. A study into the usability of the kit by spreadsheet practitioners would reinforce the value of this work.

Nixon and O'Hara (2001) underline weaknesses in spreadsheet auditing software by experimenting with auditing packages and an error seeded spreadsheet. The research found that the auditing software could detect close to 80% of the errors in the spreadsheet.

For example, the auditing software, using Panko and Halverson's (1998) error types, was found to be strong at detecting mechanical and logic errors but poor detecting omission errors.

Further, Nixon and O'Hara stress that audit software can only indicate where errors potentially lie, the effectiveness of the software is therefore still partially dependant on the skill of the auditor.

Flood and McDaid (2007) present a novel approach to debugging spreadsheets using voice recognition software. However, the results showed that debugging the spreadsheets by voice took almost twice as long and detected 15% less errors when compared to traditional spreadsheet auditing methods.

2.6.3.2 Spreadsheet control software

Spreadsheet control software is defined as software that allows the management of spreadsheet models by controlling how spreadsheets are used. There are currently two types of control mechanism, centralised and decentralised.

Centralised control mechanisms, such as Google spreadsheets, require modellers to use or download spreadsheets from a central server, modify them and then replace them. These systems keep copies of previous versions so that if a mistake is made, a rollback can be performed. Modellers who wish to use or download spreadsheets have

to log into a server and all changes to the spreadsheet are recorded providing an audit trail.

Decentralised control systems place controls on the users' PCs to monitor spreadsheet usage and modification. Typically these systems employ agent technologies that monitor spreadsheet modification and make comparisons between versions of the same spreadsheet. Any changes to the spreadsheet are recorded and attributed to a user, providing an audit trail.

"Telltable" is a centralised control mechanism, developed by Nash (Nash 2003 and Nash and Goldberg 2005). The product allows the user to track changes and rollback to previous versions. This software also has standard auditing capabilities.

Two observations can be made of centralised control mechanisms, firstly the need for investment in technology and secondly a change in 'normal' use of spreadsheets. Investment in technology may be necessary to adopt a centralised system, for example a suitable application server may need to be purchased.

'Normal', use of spreadsheets is defined as spreadsheet software residing on modellers own PC, which they open and use on their PC. A centralised system requires a hosted or downloadable spreadsheet.

One approach to utilising agent technologies in spreadsheets to reduce spreadsheet error is proposed in Thorne *et al.* (2003).

A similar approach is Baxter (2004) presents a decentralised control system that uses agents to monitor change in spreadsheets (Baxter, 2004). Agent software monitors spreadsheets on a network and once a change is recorded, two versions of the same spreadsheets are analysed for differences. A report is then generated identifying changes and by whom those changes were providing an audit trail.

The approach of Baxter (2004) does not require investment in infrastructure, or centralisation of spreadsheet software that Nash (2003) and Nash and Goldberg (2005) require. However, it does make use of decentralised agent technology which

attracts criticism. For example, Nwana and Ndumu (1999) note security as a primary concern in agent technologies, whilst other criticisms of the technology include increased loading on LAN bandwidth and unsatisfactory transaction control mechanisms. These are criticisms of agent technology and not directly Baxter (2004) although the same criticisms apply.

2.6.3.3 Alternative spreadsheet programming environments

Alternative spreadsheet programming environments are defined as non-conventional methods for programming spreadsheets. Conventional spreadsheet programming methods are defined as the traditional matrix analogy using cells and formulae, such as Microsoft Excel or Lotus 123.

Paine (2001) presents "Model Master" a tool for backward engineering and creating spreadsheets. Model Master converts spreadsheets into precise mathematical notation, the output resembling traditional computer programming code. Model master allows spreadsheet modellers to construct new spreadsheets by coding the spreadsheet in the Model Master language and then converting the language into a spreadsheet.

Paine (2005) demonstrates how modularity can be achieved in spreadsheet models via Model Master. Paine (2005) argues that modularity allows the effective management of large spreadsheet models.

In Paine (2006) and Paine (2007) Model Master is re named as "Excelsior", with new emphasis being placed on the benefits of modularity.

However, since Excelsior is similar to a traditional programming language, considering that most spreadsheet developers are non-IS professionals, they are unlikely to have programming experience. This may mean that spreadsheet modellers will encounter difficulty using such an approach without training.

2.6.4 Conclusions on error reduction literature

From sections 2.6.1.1, 2.6.1.2 and table 2.6 it seems that the most effective means of reducing errors in spreadsheets based upon the available literature is manual auditing. Individual audits find between 51% and 67% of errors, whilst team audits find 83% of errors

As discussed in section 2.6.1, applying software engineering principles to spreadsheets and forming a spreadsheet engineering discipline has the potential to significantly reduce spreadsheet errors. Evidence presented by Rajalingham *et al.* (2000), Burnett *et al.* (2001), Burnett *et al.* (2003) and Burnett *et al.* (2004), show reductions in errors that are gained by applying software engineering principles to spreadsheets.

Section 2.6.2.2 considers research on spreadsheet testing conducted by Grossman (2002), Yirsaw *et al.* (2003), Grossman and Ozluk (2004), Pryor (2004), Panko (2006) highlight the potential benefits to be gained from software engineering testing methods and the creation of a spreadsheet engineering. However, in these papers no data is included to prove the effectiveness of such approaches.

Section 2.6.3.1 concludes that auditing software has an unknown effect on reducing spreadsheet errors. Whilst the software is good a finding particular types of error, as Nixon and O'Hara (2000) stress, auditing software can only point to potential errors, deciding if they are errors and how to correct them is left to the auditor.

However, one exception to the above conclusions on auditing software is work conducted by Markus Clermont and colleagues in (Clermont *et al.* 2002, Clermont 2003, Clermont and Mittermeir 2003 and Clermont 2004). The above work demonstrates how a visualisation tool can be used to audit spreadsheets in a unique manner.

Further, in Clermont and Mittermeir (2002) the authors take a number of spreadsheets from industry and audit them using this software. The only criticism of this is that the

auditing was conducted by the authors, raising questions of usability by spreadsheet modellers themselves.

Section 2.6.3.2 considers spreadsheet control software, both the centralised and decentralised environments, offers a means of controlling spreadsheet development in an organisation rather than directly reducing errors.

These control systems offer theoretical benefits but lack hard evidence that proves the effectiveness of such software. Further, there are potential security and investment disadvantages associated with some of the approaches.

Alternative approaches to spreadsheets are considered in section 2.6.3.3. Alternative approaches to spreadsheets use precise notation to define spreadsheets as a computer language (Paine 2001, Paine 2005, Paine 2006 and Paine 2007). Whilst this approach has the potential to reduce error, there is no data to prove it.

In addition spreadsheet modellers, non-IS professionals, may have difficulty with using a language that resembles traditional programming without sufficient training.

2.7 Spreadsheet errors – A mismatch between man and machine?

As stated in section 2.5.4, human factors play a significant role in spreadsheet errors and have been largely ignored by the wider spreadsheet community. Some authors (Panko, 2007) suggest that spreadsheet errors are human errors found in spreadsheets.

The quantifiable human factors such as BER (Panko, 2005) and Cognitive load (Kruck *et al.* 2003) have been shown to have an effect on spreadsheet quality. Other aspects such as Miller's threshold (Thorne *et al.* 2004) bear on the relationship between spreadsheets and error

The effect of human factors on spreadsheets and information systems generally can be partly attributed to poor interaction between humans and computers. After listening to

the keynote speech of Ray Panko at the 2005 EuSpRIG conference where he stressed new research much come from the human factor research area, it was clear that a novel insight was required. On reflection this novel insight was inspired by the earlier work of Donald Michie and therefore became the starting point for the **novel contribution of this thesis**, as explained below.

Donald Michie in the 1980's researched the interaction of human and computers at length, although much of Michie's work predates the modern human factors research.

Much of Michie's work throughout the 1970's to 90's was concerned with Machine Learning Techniques (MLT) and the comparison of those techniques with equivalent human abilities (Michie, 1979 and Michie, 1990). He also revealed insights into the human learning process (Michie, 1982).

For example, in Michie *et al.* (1989) a comparison was made between human and machine learning on the "legality" of positions in end-game chess: King and Rook Vs King moves. Examples of illegal positions are where either the rules of the game are breached (two pieces occupy the same square) or where the game cannot proceed because of check (the white rook and the black king being on the same file).

Both the human and machine participants were presented with the same series of position scenarios and asked to classify whether the positions were legal or not. The machine learning algorithms substantially out performed the human competitors by classifying more moves correctly, the results of the experiments were: 84.2% (Machine) to 51.2% (Human) in experiment 1 and 99.0% (Machine) to 79.3% (Human) in experiment 2. Michie concluded that high performance in the tasks was almost entirely dependant on the ability of the competitor to express first-order predicate relationships. The conclusion therefore is that the machine learning algorithms out performed the human counterparts because of their superior ability to express first order predicate relationships, the very kernel of logic.

Michie *et al.* (1989) is a good example of the typical research undertaken by Michie throughout his career, this paper in particular highlights an important issue in Human

Computer Interaction – computers are significantly superior at manipulating logic in comparison to humans.

The body of Michie's research suggests that the roles of machine and human in interaction did not exploit either's strengths. In that vein, we argue that spreadsheet errors are mainly attributed poor interaction between humans and computers.

Consider the way in which a user interacts with a computer to create a spreadsheet model, such as a business problem, there are two fundamental processes.

The first element is matching patterns in real–world examples and realising trends in those patterns that form some rule or judgement. This allows the modeller to interpret and rationalise the problem and make rules that operate in that problem.

The second is the manipulation of mathematics and logic to represent that system accurately, which could be via a spreadsheet or another tool.

Now if we consider table 2.7, the natural strengths of the average human and the conventional computer, some discrepancies arise.

	Pattern matching	Generating real-world examples	Manipulating mathematics	Logical deduction
Human	Strong	Strong	?	?
Conventional Computer	Weak	Weak	Strong	Strong

Table 2.7 Strengths and weaknesses in conventional computers and humans

From this table we suggest that humans are strong at generating real-world examples and pattern matching but weak at mathematical manipulation and logical deduction. Conversely, computers are strong at manipulating mathematics and logical deduction but weak at generating real-world examples and pattern matching.

In the current spreadsheet paradigm, strain is placed on the natural weakness of the human, logical deduction and manipulating mathematics, i.e. thinking up formulae to satisfy a problem (see the circled section on table 2.7). Further the strengths of the

human (pattern matching and generating real-world examples) are not exploited by the current spreadsheet paradigm.

A potentially more beneficial paradigm would be to play on the natural strengths of the human and the conventional computer. In this new paradigm, the human would pattern match and generate real world examples, the computer would use its ability of mathematical manipulation and logical deduction to build a model from the examples provided by the user, see the circled sections on table 2.8. This idea is explored at length in Thorne and Ball (2009).

	Pattern matching	Generating real-world examples	Manipulating mathematics	Logical deduction
Human	Strong	Strong	?	?
Conventional Computer	Weak	Weak	Strong	Strong

Table 2.8 Proposed methods of interaction

2.8 Opportunities for novel research – example-giving

One novel alternative approach to reduce spreadsheet error would be to use machine learning techniques in the creation of spreadsheet models (Thorne *et al.* 2004)

The novel approach would use real world examples provided by the user. i.e. the user would think up examples of input and output for a given problem.

The computer, using a machine learning technique, would deduce the mathematics and logic of the examples and generate a 'model' to reflect the examples.

It is thought that this approach could reduce spreadsheet error by reducing the impact of certain human factors during development.

For example, the cognitive load of thinking up examples should be significantly lower than that of spreadsheet modelling, see table 2.5.

However, some human factors may still be present. For example Base Error Rate (BER) is present in all human activities whether it be spreadsheet modelling or thinking up examples.

2.9 Summary of literature on spreadsheet errors

Considering the literature review on spreadsheet error and relevant human factors, several issues relating to spreadsheet error are particularly important. The following issues are significant sources of what might be regarded as the "noise" of spreadsheet error: BER, Bias and Poor programming.

Conclusions to the literature review on the topics of spreadsheet errors, relevant human factors and error reduction methods are contained in sections 2.4.5, 2.5.3 and 2.6.4 respectively.

In summary section 2.3 discusses spreadsheet error rates, and also notes that taxonomies of spreadsheet error are similar to taxonomies of human error. Further, there is a significant link between spreadsheet error and human factors as suggested by Panko (2007) which warrants investigation.

Section 2.5.4 concludes that there is a quantifiable relationship between particular human factors and spreadsheet errors. Further, any novel method must take account of this relationship in order to mitigate the effect of human factors

Section 2.6.4 concludes that spreadsheet engineering methods have the potential to reduce errors but are yet to be widely adopted. Spreadsheet auditing software is good at detecting particular types of errors but poor at others, in any case the actual correction of errors is left to the auditor. Alternative spreadsheet software shows some promise but requires a greater level of skill from the spreadsheet modeller than they are likely to possess.

Section 2.7 discussed how human factors relate to spreadsheet errors and how together with the work of Michie this provided and inspired the foundations of the novel contribution of this thesis.

The above conclusions draw the literature review to a close and thus satisfy the first objective of this thesis:

> *"Undertake a literature review of relevant topics within the field of spreadsheet error research"*

The conlusions of the literature review establishes the need for alternative paradigms to be considered as objective 2 states:

> *"Based upon the literature review, consider an alternative modelling technique for the reduction of error in decision support spreadsheets"*

3.0 Investigating the feasibility of example-giving

3.1 Overview of the chapter

The structure of this chapter is as follows, section 3.2 introduces the chapter, briefly explaining the purpose of chapter 3. Section 3.3 outlines the novel approach, Example Driven Modelling (EDM) and explains the need for feasibility testing. Sections 3.4 and 3.5 introduce and deal with the design aspects of the feasibility experiment. Section 3.6 presents the summary statistics generated from the results of the experiment. Section 3.7 determines if relationships present in the summary statistics are statistically significant using a number of significance tests. Section 3.8 draws conclusions on the experimentation and considers the limitations of the experiment. Section 3.9 assesses what impact the conclusions and limitations have on the novel approach and outlines further work for the next chapter. Finally section 3.10 provides a summary on the work contained in this chapter.

3.1.1 The problem domain

The problem domain is defined by the objective 2
Section 1.4.3 objective 2 stated:

Based upon the literature review, consider an alternative modelling technique for the reduction of error in decision support spreadsheets

The purpose of this chapter is to establish if the **alternative modelling technique for the reduction of error** identified in the literature review, **"example-giving"** (see section 2.8), is feasible.

The example-giving technique is suited to classification problems (Thorne and Ball, 2006b). That is to say where problems are classified according to some defined **logic** as found in **decision support spreadsheets** (Thorne and Ball, 2006b). However the scope of the technique does **not** include purely mathematical models such as financial spreadsheets (Thorne and Ball, 2006b) because the number of examples required for any non-trivial problem would quickly become NP-complete.

3.2 Introduction

The purpose of this chapter is to establish if the novel approach identified in the literature review, **"example-giving"** (see section 2.8), is feasible. The feasibility is determined through an experiment which compares traditional spreadsheet modelling techniques with the novel approach. Using experimentation it is possible to establish if example-giving offers a significant advantage over traditional methods.

The example-giving process is the basis for a novel approach, the novel approach uses examples (attribute classifications) given by the modeller to build a model of those examples which can be applied to new unseen examples. This novel approach is coined Example Driven Modelling (EDM)

3.3 Example Driven Modelling

Example Driven Modelling (EDM) uses example attribute classifications, provided by the user, to compute the function of those examples into a generalised model via a machine learning technique.

Figure 3.1 shows the EDM process from start to end. Firstly the user would have to provide example attribute classifications for the problem they wish to model. The examples are then formatted into a data set and fed through a learning algorithm. The algorithm learns from the example data provided, which results in a general model. The general model is then able to generalise to new unseen examples in the problem domain.

Figure 3.1 Example Driven Modelling

This approach eliminates the need for the user to produce formulae, the user only gives example data for the problem they wish to model. This therefore eliminates errors in constructing formulae since the user is no longer required to produce them.

The burden of calculation is placed on the computer, which using a machine learning algorithm, computes the function of the examples. As the literature suggests, this may be a more effective use of human and computer strengths (Michie *et al.*, 1989)

However, the feasibility of EDM has not been established, therefore the next section explores how the feasibility of EDM can be evaluated using a suitable approach.

3.4 Investigating the feasibility of giving examples

In order to objectively determine the usefulness of example-giving, a means to measure example giving against the traditional alternative, spreadsheet modelling must be devised.

The aims of the feasibility study are

1. To determine if example-giving gives an advantage in terms of accuracy, ease of use and participant overconfidence when compared to traditional spreadsheet modelling.
2. Determine if the effects observed in the first aim are statistically significant.

Research methodology and research approach can be viewed in several different ways. Broadly there are two divisions in research approaches, from a philosophical point of view these are Phenomenology and Positivism.

3.4.1 Research methodology

The research methodology incorporates five different research components that together form a coherent approach to conducting research. The five components are: Research philosophy; Research approach; Research strategy; Time horizons and Data collection methods.

Careful consideration of the research methodology results in a well formed comprehensive practical approach to answering the research question.

The following sections have been produced by considering research methodology design texts such as Saunders *et al.* (2007) and Maylor and Blackmon (2005). Further, research methodologies employed in spreadsheet literature are also considered.

A useful overview of the research process is the '*Research process onion'* presented by Saunders *et al.* (2007), see figure 3.2.

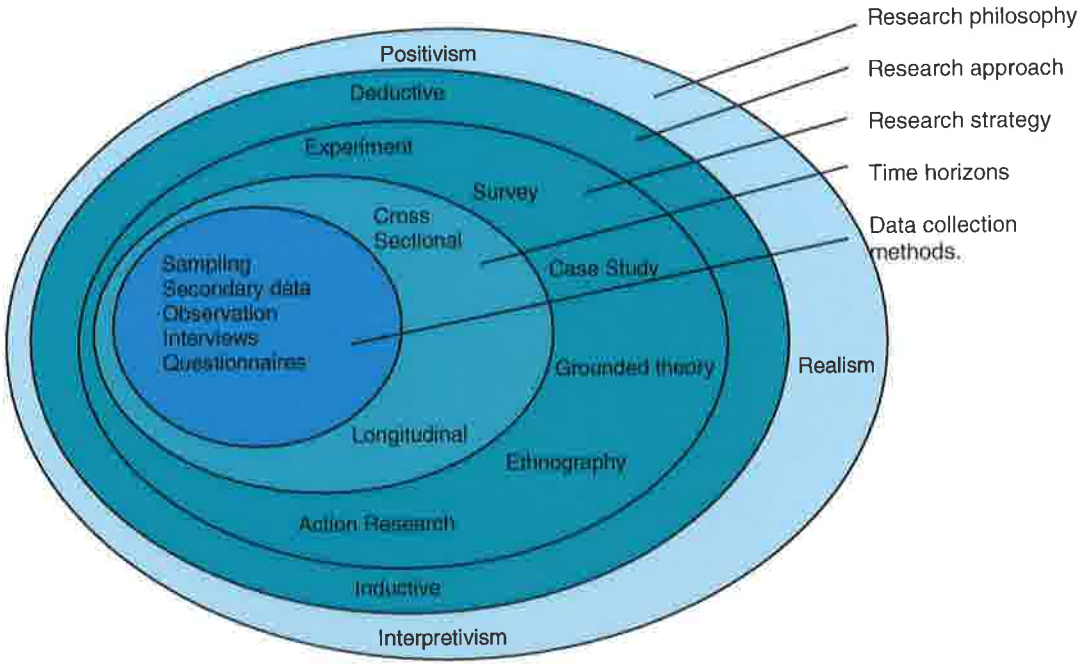

Figure 3.2 Research process onion, adapted from Saunders *et al.* (2007)

The research onion shows the five components that together form a research methodology and offers some guidance on the relationships that exist between them.

The research onion is considered systematically from the outer ring to the centre core therefore the first consideration is research philosophy.

3.4.2 Research philosophy

Selection of research methodology is influenced by the researcher's philosophical viewpoint, i.e. one is likely to adopt research methods that are complimentary considering the research philosophy of the individual.

However, this does not mean that phenomenological, positivist and realist methods cannot be combined; in fact combination of methods is common place and can be beneficial as noted by both Saunders *et al.* (2007) and Maylor and Blackmon (2005).

Interpretivism (phenomenology) is a philosophy that argues phenomena can only be truly experienced by subjective immersive study. A phenomenologist research

methodology is typically inductive, qualitative and is influenced by the experience of the phenomena by the researcher.

Positivism is a philosophy that argues phenomena can be objectively measured by the researcher to understand the nature of the phenomena from an external view point. A positivist research methodology is typically deductive, quantitative and is not influenced by the researchers own experience of the phenomena.

According to Saunders *et al.* (2007) realism is based upon the belief that reality exists independent of human thoughts or beliefs but acknowledges that collective opinion influences how others think. In that vein realism has similarities to both positivism and phenomenology.

Selection of a research philosophy is more a personal choice than an evaluation of which is *better*. Although some situations lend themselves to particular philosophies, more often than not, one could take any philosophical stance on a particular matter.

3.4.3 Research approach

Moving from the outer ring inwards, the next ring deals with 'research approach'. According to the research onion, a research approach can either be deductive, testing rules or theory by collecting data or inductive, generalising rules or theories from the gathered data.

Typically but not exclusively, deductive approaches are used in positivist research and inductive approaches are used in phenomenologist research.

Examples of deductive research approach might be hypothesis or theory testing in experimentation. In such an example the researcher may wish to confirm or challenge an existing or novel rule or theory by systematically testing the conditions that make the rule or theory true i.e. the data collection leads the theory.

An example of inductive research might be the triangulation of opinions gathered from interviewing to form a rule or judgement on a particular event or object. An inductive research approach synthesises rules by triangulating information gathered from the interviewees, i.e. the theory is led by the data collection.

Once the research approach has been considered, the next step is to decide on an appropriate research strategy.

3.4.4 Research strategy

According to figure 3.2, the available research strategies are: Action research; Ethnography; Grounded theory; Case study, survey and experiment.

The research strategy is the practical method to collect data which is heavily influenced by the research approach and philosophy, i.e. a researcher is more likely to choose action research if the research strategy is inductive and if the research philosophy is interpretivism (phenomenology).

3.4.4.1 Action research

According to Maylor and Blackmon (2005) action research is the process of changing some aspect of an organisation with the researcher taking an active role in the change. So in other words the researcher becomes part of the organisation and together with the other members of the organisation, change is effected for the gain of the whole organisation.

Saunders *et al.* (2007) stress that action research is different to other forms of applied research because the emphasis is on change and not observation, as highlighted on the following quote:

> *'The purpose of action research and discourse is not just to describe, understand and explain the world but also to change it'*
>
> Coghlan and Brannick (2001)

Maylor and Blackmon (2005) suggest that action research is more associated with research with a social component such as *'equality, fairness or the environment'*

Action research is associated more with a phenomenological research philosophy although not exclusively.

Considering the aims of the feasibility study in section 3.4, it is impossible to answer using action research as a research strategy since action research demands an inductive approach and the aim requires a deductive answer.

3.4.4.2 Ethnography

Saunders *et al.* (2007) define the purpose of ethnography as to: *'interpret the social world the research subjects inhabit in the way in which they interpret it'*

In other words ethnography is immersive study in which the researcher seeks to portray the perceived reality of the research subjects, such as workers in an organisation, by participant observation.

The researcher joins the organisation and immerses themselves in the social reality of the organisation to provide a more accurate reflection of issues within in the organisation over a period of time.

Ethnography is distinguished from action research since ethnography does not seek to implement change but to observe and report the subject's perceived reality.

Ethnography demands an inductive approach and like action research is more associated with phenomenology and realism than a positivist philosophy.

For the same reason that action research cannot be applied to this thesis, ethnography cannot be applied since it demands an inductive research approach.

3.4.4.3 Grounded theory

Grounded theory as defined by Saunders *et al.* (2007) as *'theory building'* through a combination of inductive and deductive methods. Typically a researcher using grounded theory collects data without an initial theoretical framework.

Once the data is collected and analysed, potential theories that are identified in the collected data by the researcher and are tested deductively by further data collection. This further data collection may reveal new trends leading the researcher to adapt theories and collect data again to test these new ideas.

Grounded theory is more associated with realism than either phenomenology or positivism although it does use both inductive and deductive methods.

For the purposes of this feasibility study grounded theory could be applied but is not well suited because it uses mostly an inductive approach.

3.4.4.4 Case study

Creswell (1994) defines a case study as:

> *'A single, bounded entity, studied in detail with a variety of methods, over an extended period'*

In other words, case study research is a longitudinal study using multiple data collection methods.

Maylor and Blackmon (2005) add that case study research is bound by a *'social unit'*, *'such as a person, a company, a situation or whatever'*. So the focus of case study research is the research subject whether that is an organisation, a person an artefact or a process.

Saunders *et al.* (2007) suggest that relevant data collection methods include questionnaires, interviews observation and documentary analysis. Further the use of multiple methods is referred to as 'empirical' research.

Since case study research can use multiple data collection methods, it is not bound to an inductive or deductive research approach. In that vein it is not particularly associated with any one research philosophy since combinations of data collection methods from phenomenology, realism and positivism are possible.

A case study could be used to determine the feasibility of example-giving, however a case study approach is not suited to making an 'objective' comparison of two methods. Further, case study research is typically qualitative rather than the detailed quantitative analysis necessary to satisfy the aim of the feasibility study.

3.4.4.5 Survey

According to Maylor and Blackmon (2005) a survey is a method of asking research subjects a range of questions from a distance usually from a distance to economically gather data typically through a questionnaire.

However, a survey may also be implemented by interviewing research subjects, using a structured interview approach.

Saunders *et al.* (2007) suggest that surveys are an economical way to gather data from a large proportion of the target population but do not offer the depth of information that other techniques do.

Further Saunders *et al.* (2007) stresses that whilst questionnaires a simple and quick way of gathering data, the opportunity to produce a poorly designed a questionnaire is considerable.

Surveys lend themselves to a deductive research approach and are a common positivist research strategy.

A survey could be used to gather opinions on example-giving, i.e. a questionnaire could be sent out asking participants if they understood or liked the process of

example-giving. However, a survey cannot facilitate an objective comparison of practical techniques alone.

3.4.4.6 Experiment

According to Shadish *et al.* (2002) an experiment is defined as:

'A test under controlled conditions that is made to demonstrate a known truth, examine the validity of a hypothesis, or determine the efficiency of something previously untried'

The above definition asserts that experimentation can be used to prove or disprove a truth but can also be used to test a novel hypothesis.

Central to experimentation is the notion of causality or cause and effect. Cause is defined by Shadish *et al.* (2002) as '*The producer of an effect, result, or consequence'*.

So the point of experimentation is to establish the causal relationship between one variable and another.

Experimentation is firmly rooted in the understanding of natural sciences although according to Saunders *et al.* (2007) social sciences, particularly psychology employs experimentation extensively.

Experimentation is strongly associated with both a positivist research philosophy and a deductive research approach.

Experimentation is more suited to testing the feasibility of example-giving than any other available research strategies.

Firstly comparison of example-giving and a traditional spreadsheet approach fits as a treatment and control group which is integral to experiment design. This allows

'objective' comparison between the treatment (example-giving) and control (traditional spreadsheet modelling) under controlled conditions.

Controlled conditions ensure that the effects observed in the results are due to the treatment and not some other phenomena.

Secondly detailed quantitative analysis of experimentation results is a standard feature of experimentation (Shadish *et al.* 2002) and therefore can provide the detail needed to answer aim of the feasibility test.

3.4.5 Research methodologies used in spreadsheet research

Most spreadsheet research is conducted from a positivist stance possibly because of the technical nature of the subject.

A range of research strategies are used in spreadsheet literature, although some are more common than others.

The use of experimentation and quasi-experimentation in spreadsheet research is extensive, some examples include: (Hicks and Panko 1995, Javrin and Morrison 1996, Panko and Halverson 1998, Javrin and Morrison 2000, Howe and Simkin 2006). This is especially true when testing novel ideas (Rust *et al.* 2006, Vemula *et al.* 2006, Flood and McDaid 2007, Bishop and McDaid, 2007)

Surveys via questionnaire and interview are also common, a few examples include: (Cragg and King 1993, Davies and Ikin 1987, Floyd *et al.* 1995, Hall 1996, Schultheis & Sumner 1994, Purser and Chadwick 2006, SERP 2006, Baker *et al.* 2006)

One could interpret attempts to create best practice as a having grounded theory as a research strategy since typically refining best practice is a recursive process with ideas changing with the experience of the researchers. Good examples of this are Grossman (2002), Grossman and Ozluk (2004) and Colver (2004).

The use of case studies as a research strategy is much less common, two examples of case study research are Gosling (2002) and Fernandez (2003).

There appears to be no instances of ethnography or action research as a research strategy in spreadsheet error research.

3.4.6 Conclusions on research methodology

The chosen research methodology is as follows:

Firstly the preferred research philosophy is positivism using a deductive research approach.

Secondly the most appropriate research strategy is experimentation since it offers an objective controlled means to test example-giving against traditional spreadsheet modelling. Further, summary statistics generated from the results can be tested for statistical significance.

The timescale of an experiment is always cross-sectional, i.e. a singular moment in time. The down side of this is that the experiment can only be considered as a snap-shot of what was true, however careful experiment design allows one to generalise the effect observed outside of this single snap-shot.

Data collection methods were selected and designed in accordance with experimental design texts such as Shadish *et al.* (2002) and Campbell and Stanley (1963). These design details can be found in section 3.5, feasibility experiment design.

3.5 Feasibility experiment design

To investigate if giving examples works in practice, an experiment was designed to compare traditional spreadsheet modelling techniques and the novel approach of giving examples. The first group, the "treatment" group, were required to give example data to complete the tasks. The other group, the control group, were given the same tasks to complete using a spreadsheet application.

The experiment into feasibility was designed in accordance guidelines cited by Shadish *et al.* (2002) and Campbell and Stanley (1963). Also, published work using experimental methodologies in spreadsheet research were considered (Hicks and Panko 1995, Javrin and Morrison 1996, Panko and Halverson 1998, Javrin and Morrison 2000, Howe and Simkin 2006)

3.5.1 Experiment aims

The main aim of the experiment was to establish experimentally within an academic environment, using postgraduate students:

1. The relationship between error and task complexity using
    1. Spreadsheet modelling techniques
    2. Example-giving
2. The (hypothesised) superiority of example-giving over traditional spreadsheet modelling.
3. A more satisfactory statistical measure of overconfidence.
4. The relationship between previous spreadsheet experience and accuracy for both traditional spreadsheet modelling and example giving

From these experimental aim and objectives, we determined the feasibility of example-giving via three performance indicators

1. The participants understanding of example-giving, i.e. whether users generated valid examples given a problem scenario
2. The accuracy of the examples provided by the participants, i.e. the error rate for examples provided by participants
3. The comparative error rate when compared to traditional modelling, i.e. the example-giving error rate compared to that of traditional modelling.

3.5.2 Experimental design

The experimental model chosen to evaluate the aims of the experiment was: *'The classic Randomised two group no post test design'* as discussed by Shadish *et al.* (2002). Figure 3.3 shows the standard design of such experiments.

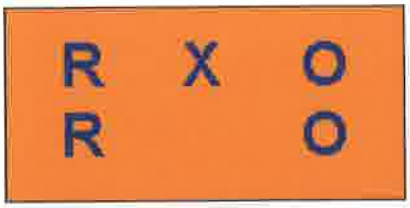

Figure 3.3 Randomised two group no post test (Shadish *et al.* 2002)

The diagram shows the two randomised (R) groups, the treatment group (X), The control group (which is left blank) and the two outcomes (O).

In this case the control group receive 'standard' treatment, i.e. they develop spreadsheet formulae using the constructs and syntax in a spreadsheet application, such as excel. The treatment group received the novel approach, this allowed relative comparison between the control and treatment groups.

3.5.3 Sampling

This sampling for this experiment is a cluster random sample as described by Shadish *et al.* (2002) and Saunders *et al.* (2007). Cluster sampling identifies a suitable cluster of participants and then randomly selects from within that group.

Considering similar development experiments in spreadsheets, postgraduate masters students were selected as an appropriate cluster (Hicks and Panko 1995, Javrin and Morrison 1996, Panko and Halverson 1998).

No particular incentive was given to the participants to attend other than assisting a staff member in some research. At a later date the results were presented and the purpose of the research explained to participants in a lecture.

Participants were invited to attend a session arranged for the experiment. Upon arriving participants were divided into two groups, the control and treatment groups. This division was based upon the order in which they attended the arranged session, as participants arrived they were alternately assigned to treatment and control groups.

The treatment and control groups completed their respective tasks in separate adjacent rooms with supervision from experimenter and a helper. This supervision was to merely ensure that no collusion took place and to answer simple questions (not relating to the completion of the tasks) the participants may have had. The participants were given as much time as needed to complete the tasks in their respective rooms. Once complete, participants submitted work to the appropriate supervisory member.

The total number of participants who attended was 49, 25 in the treatment group and 24 in the control group. However, one participant of the control groups supplied a corrupt spreadsheet obviously this meant that this particular result could not be included. The total number of participants therefore was 23 and 25 for the control and treatment groups respectively.

3.5.4 Research materials

The research materials for this experiment comprised a control group and treatment group pack handed to the respective participants. The details of each pack are listed below.

Control group:

1. Questionnaire 1 (Age, Sex, Spreadsheet experience)
2. Control group materials (Spreadsheet tasks)
3. Questionnaire 2 (Self evaluation of performance)

Treatment group:

1. Questionnaire 1 (Age, Sex, Spreadsheet experience)
2. Treatment group materials (EDM tasks)
3. Questionnaire 2 (Self evaluation of performance)

Questionnaire 1 gathered information such as age, sex, experience, number of years using spreadsheets, and a personal rating of their skill. Questionnaire 1 was completed first, before the participants started the tasks. The point of questionnaire 1 is to gather demographic information and to determine the experience of spreadsheet use for a participant.

Once questionnaire 1 was completed, the participants started the tasks for the group they were assigned to (control or treatment). The participants from the treatment and control groups were required to provide valid solutions to five progressively more complex scenarios.

The manner in which the groups completed the tasks differed, the control group produced formulae in a spreadsheet using the syntax and functionality of the application (Microsoft Excel). The treatment group produced example attribute classifications for each task.

After completing the tasks as best they could questionnaire 2 was completed. This questionnaire gathered information on the participant's perception of their own performance, i.e. they were asked how difficult they felt each task was and then asked to indicate how confident they were that the provided answers were correct.

3.5.5 Experiment tasks

The experiment tasks were designed to be progressively more difficult, requiring progressively more complex answers from treatment and control groups.

The control group submitted answers created using Microsoft Excel, the treatment group submitted attribute classifications written on paper.

3.5.5.1 Example Task

For the control, group task 1 was to create a formula that could give a grade (Pass or Fail) based upon a single mark (Exam mark). The formula was required to distinguish between pass and fail, where fail was < 40 and pass was >= 40.

For the same task, task 1, the treatment group were required to give attribute classifications (examples) of the two classifications in the model, Pass and Fail.

The tasks were designed to be increasingly more complex, table 3.1 contains details of each task given to the treatment and control groups. Appendix C and D contain the all the materials distributed to the Treatment and Control groups respectively.

Tasks/Group	Treatment group	Control group
Task 1	2 attribute classifications, An example of each class	1 value, 2 classes, <40 Fail, >= 40 Pass
Task 2	4 Attribute classifications, 2 examples of each class	2 values averaged, 2 classes, <40 Fail, >=40 Pass
Task 3	4 Attribute classifications, 2 examples of each class	2 values averaged where both >= 40 for min Pass, 2 classes Pass and Fail
Task 4	8 attribute classifications, 2 examples of each class	2 Values, averaged, 3 classes, < 40 Fail, >= 40 Pass, >= 55 Merit, >= 70 Distinction
Task 5	8 attribute classifications, 2 examples of each class.	2 Values, averaged, Both must fall in class range to award class, 3 classes, < 40 Fail, >= 40 Pass, >= 55 Merit, >= 70 Distinction

Table 3.1 Control and Treatment group task specification

3.5.5.2 Marking the control group

Determining the mark of participants in the control group was based upon whether the answer provided was a valid formula in Excel and whether the formula satisfied the specification in the task. If the formula fulfilled both criteria, it was deemed as correct, otherwise it was incorrect.

3.5.5.3 Marking the treatment group

Determining the mark of the participants in the treatment group was based upon the whether the attribute classifications were valid and whether the attribute classifications provided satisfied the specification of the problem.

3.5.6 Conclusions on experimental design

The conclusions on the experimental design are as follows:

The experiment follows the randomised two group no post test design as described by Shadish *et al.* (2002), see figure 3.3.

The sampling approach was a clustered random sample as described by Shadish *et al.* (2002) and Campbell and Stanley (1963). The identified cluster was Master's students, a cluster used in other experimental spreadsheet studies (Hicks and Panko 1995, Javrin and Morrison 1996, Panko and Halverson 1998).

Research materials for the control and treatment groups were as follows:

Control group:

1. Questionnaire 1 (Age, Sex, Spreadsheet experience)
2. Control group materials (Spreadsheet tasks)
3. Questionnaire 2 (Self evaluation of performance)

Treatment group:

1. Questionnaire 1 (Age, Sex, Spreadsheet experience)
2. Treatment group materials (EDM tasks)
3. Questionnaire 2 (Self evaluation of performance)

The experiment tasks for the control group and the treatment group are described in table 3.1. The experiment tasks for both groups are increasingly difficult.

Marking the answers provided by the control group was achieved by evaluating whether the formula provided was syntactically valid and if the formula satisfies the specification of the task

Marking the answers provided by the treatment group was achieved by evaluating whether the answers provided were valid attribute classifications and if the attribute classifications satisfied the specification of the task.

Marks in both groups were dichotomous, i.e. answers provided were either correct or incorrect.

3.6 Experimentation summary statistics

This section contains the summary statistics generated from the collected data. These summary statistics deal with accuracy of the two groups, relative experience of the two groups and the confidence calculations for the two groups.

3.6.1 Accuracy

By comparing accuracy results gained from both the treatment and control groups, it is evident that the treatment group were more accurate than the control group. See Figure 3.4

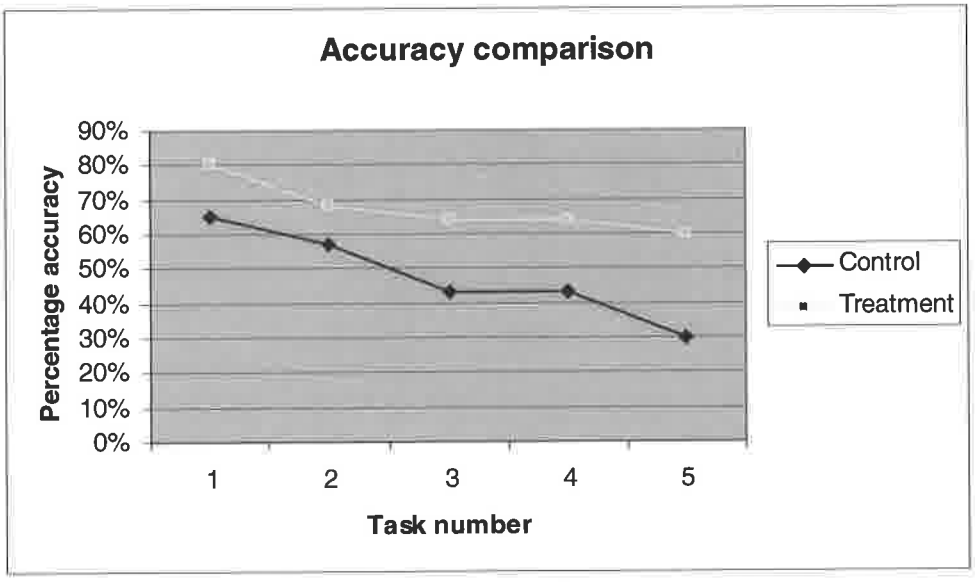

Figure 3.4 Relative accuracy between Control and Treatment groups

As can be seen in figure 3.4, the treatment task accuracy ranges between 80 and 60 percent, the control group accuracy ranges between 65 and 30 percent. So comparatively, producing examples is more accurate than producing formulae.

3.6.2 Experience and accuracy

Data regarding experience was collected through three questions:

1. "How do you rate yourself as a spreadsheet developer?" (figure 3.5)
2. "How many years have you been using spreadsheets?" (figure 3.6)
3. "What formal training have you had in spreadsheets?" (figure 3.7)

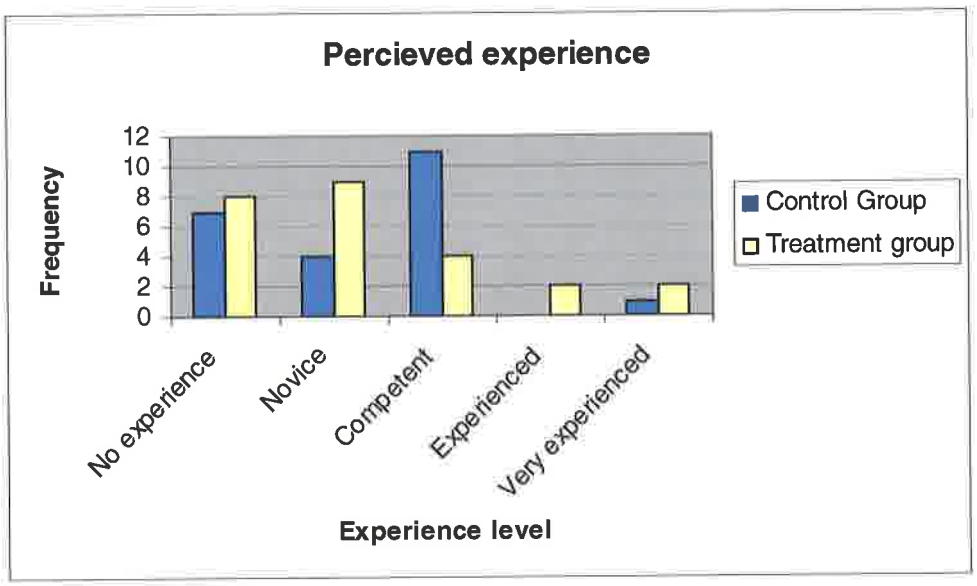

Figure 3.5 Answers to "how do you rate yourself as a spreadsheet developer?"

Figure 3.5 shows the perceived experience levels for the treatment and control group are approximately the same.

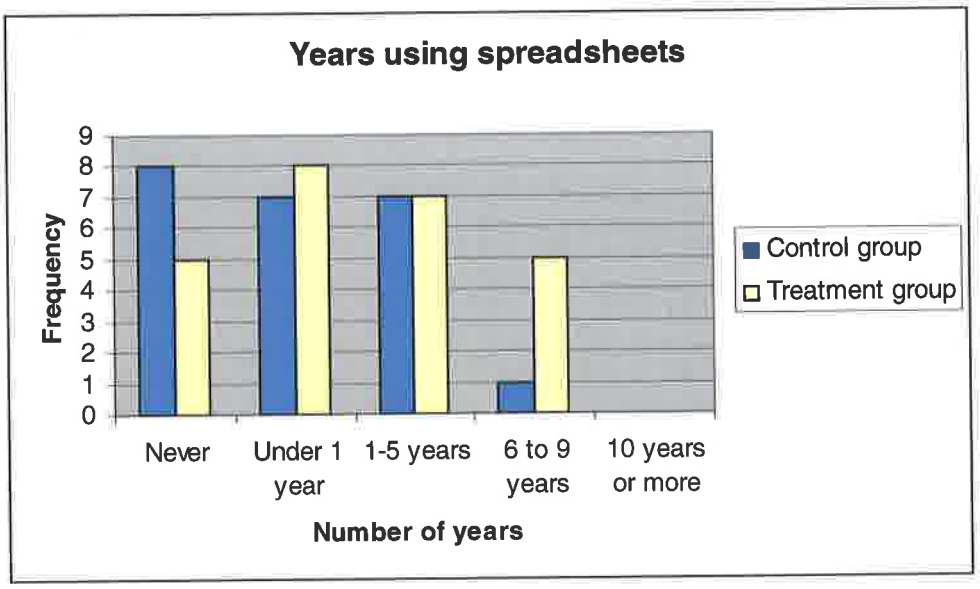

Figure 3.6 Answers to "How many years have you been using spreadsheets?"

Figure 3.6 shows that experience, in terms of years using spreadsheets, between the treatment and control groups. The frequencies perhaps show some experience bias towards the treatment group. However the statistical significance of this is considered later on. See appendices C and D for the questionnaire distributed to participants.

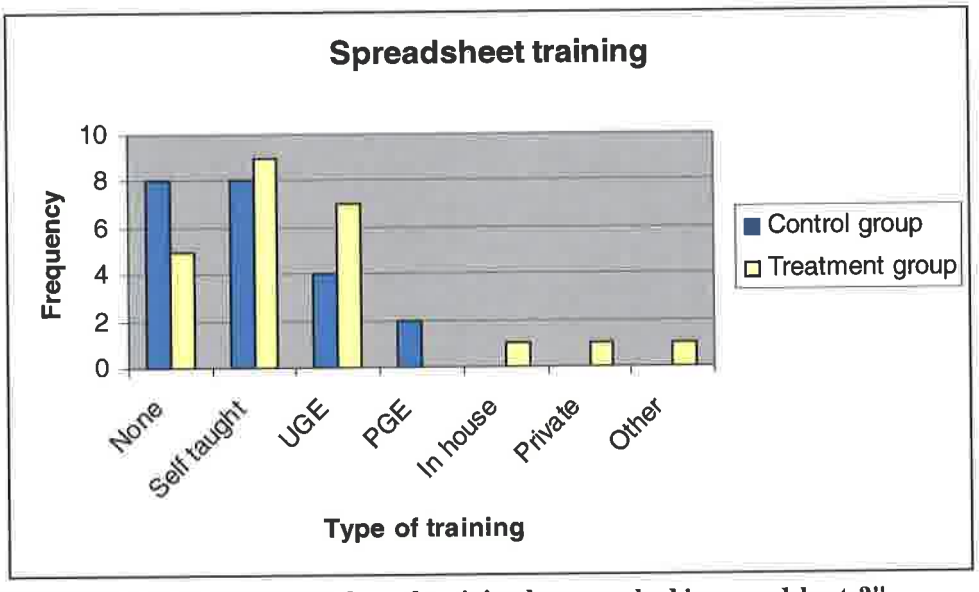

Figure 3.7 Answers to "What formal training have you had in spreadsheets?"

Figure 3.7 shows what formal training (or otherwise) the participants of the experiment had received. Predictably there are a high number of participants stating that they are self taught.

In order to understand the relationship between experience and accuracy, experience levels in figures 3.6 and 3.7 are plotted against the respective accuracy by participant.

In figure 3.8, participants have been dived into categories of No experience (if they answered "No experience") and some experience (if they answered anything other than "No experience")

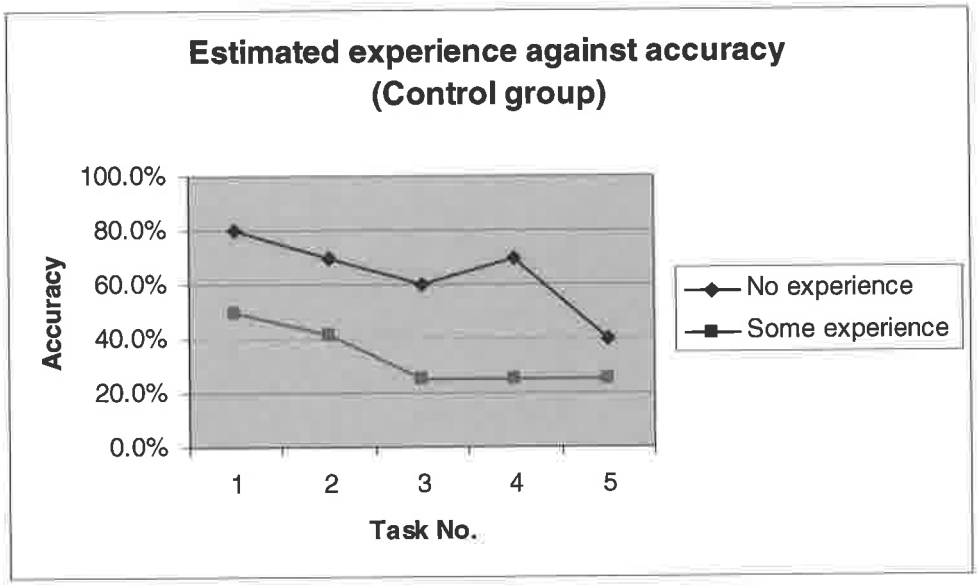

Figure 3.8 Estimated experience against accuracy

The data in figure 3.8 shows that participants who answered "no experience" were more accurate than those who answered "some experience". This is an unusual result since one would assume that the more experience the more accurate an individual.

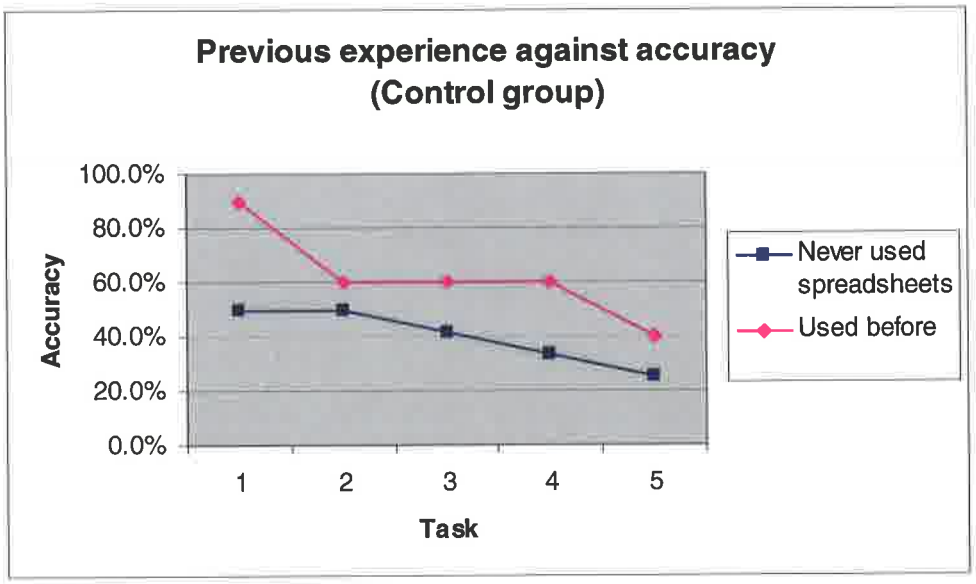

Figure 3.9 Previous experience against accuracy

Figure 3.9 compares the response to the question "How many years have you been using spreadsheets?" and accuracy for the control group. The results are categorised into those who answered "Never used them before" and "used them before". This data shows that participants who indicated they had used spreadsheets before were more accurate than those who had never used them before.

This relationship is contradictory to that in figure 3.8, in fact when the individual responses were examined in more detail, some participants had indicated that they considered themselves experienced but when questioned over the number of years using spreadsheets, they said they had never used them before. This might suggest that they misunderstood the question or have misjudged their own experience.

In the treatment group the results show that participants who answered "some experience" were more accurate than those who indicated they had no experience.

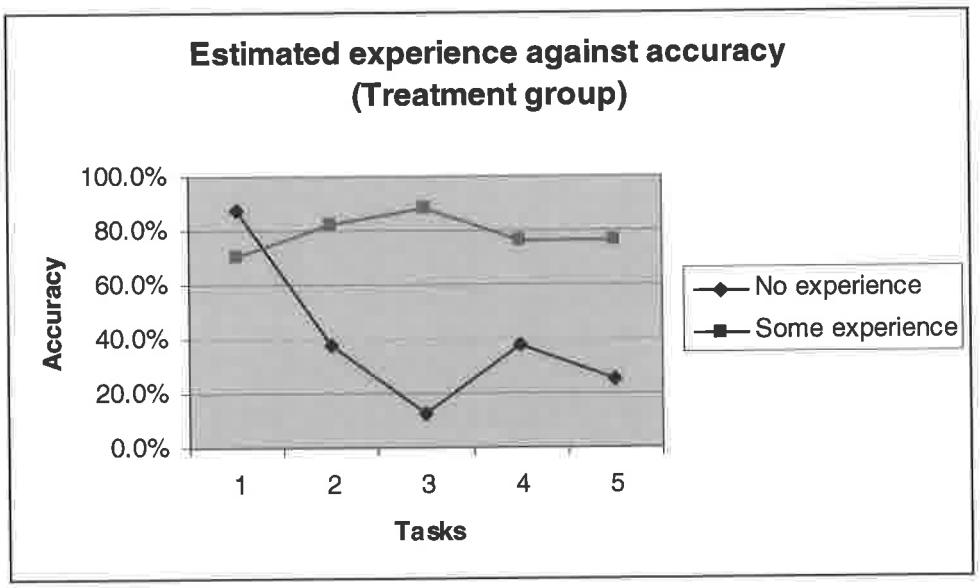

Figure 3.10 Estimated experience and accuracy (Treatment group)

Figure 3.10 compares estimated experience and accuracy in the treatment group, unlike the control group results, figure 3.10 suggests that the more experienced the participant the more accurate they were.

However, the experience question relates to spreadsheets, not giving examples. So the more experienced with spreadsheets the participant is, the more accurate at giving examples.

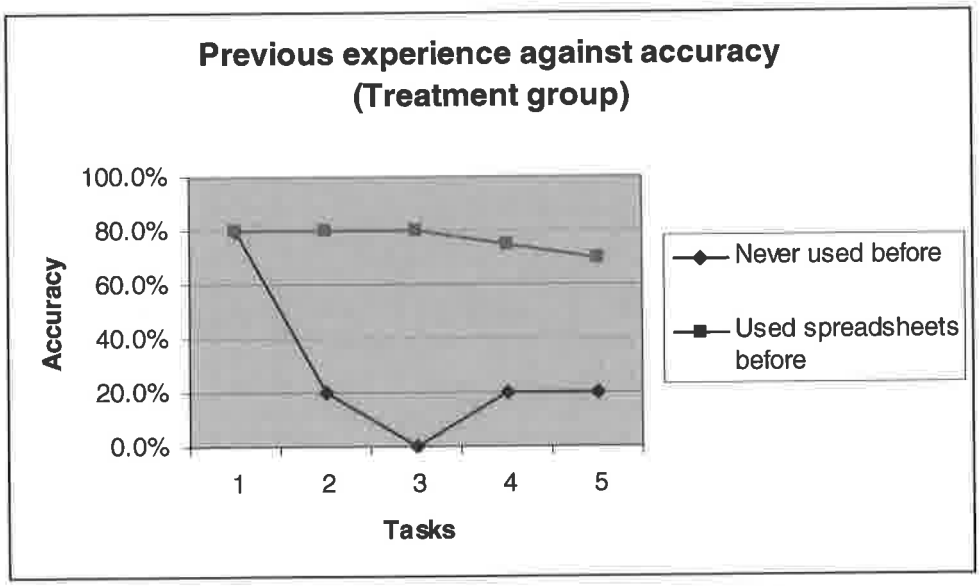

Figure 3.11 Previous experience against accuracy (Treatment group)

The data in figure 3.11 supports the data in figure 3.10, this graph shows experience as years with participant responses divided into those who stated they had never used spreadsheets before and otherwise. So those participants who indicated they had used spreadsheets before were more accurate in giving examples than those who had no experience.

3.6.3 Confidence

Since there was no standardised method for measuring confidence (over or under) in spreadsheet modelling a method was devised to this end. Thorne *et al.* (2004) introduced the "confidence ratio" which calculates the ratio between perceived performance (how well the participant believed they did with respect to the difficulty of each task) and actual performance (how well the participant actually did in each task).

Hence, the ratio provides a measure of confidence, with > 1 implying over confidence, < 1 implying under confidence, and equal to 1 implying sufficient confidence, i.e. accurately assessing their performance. The confidence ratio is calculated using Ratio Perceived Success Rate (RPSR) and Actual Success Rate (ASR)

The RPSR, seen in the nominator of equation 3.1, is derived using responses obtained from the evaluation questionnaire (see Appendices C and D) which participants were required to complete after finishing all 5 tasks. Within this questionnaire, participants were asked for their opinion on 2 measures of significance, performance in task (i.e. how successfully did the individual believe they performed) and difficulty of task (i.e. how difficult did the individual find the task), for each of the 5 tasks completed. The responses were translated into Completeness and Difficulty scores, with each score ranging from 1 to 5, defined as follows:

Completeness - Did you successfully complete the task?

1 - No

2 - Probably not

3 - Do not know

4 - Probably

5 - Yes

Difficulty - How difficult was the task?

1 - Very hard

2 - Hard

3 - Average

4 - Easy

5 - Very easy

The two scores (Completeness and Difficulty) are then combined to produce an overall measure of perceived confidence (RPSR), as seen below in Equation 3.1. Integral to this interpretation of RPSR, the scores are weighted evenly (A = B = 0.5). That is, the coefficients of the scores are equal. Note, however, the option to change the focus in this relationship by varying the values of the coefficients A and B (where A + B = 1).

$$Ratio\ Perceived\ Success\ Rate\ (RPSR) = (A*Completeness) + (B*Difficulty)$$

Equation 3.1 RPSR formula

Where

Completeness = completeness score obtained from questionnaire

Difficulty = Difficulty score obtained from questionnaire

A = weight of confidence score in RPSR relationship, set to 0.5

B = weight of difficulty score in RPSR relationship, set to 0.5

For the denominator of the Confidence Ratio in Equation 3.2 (ASR), we let x be the number of errors made in each task and let F(x) be the actual mark or result given for the task. Then, the distribution of F(x) is defined as follows:

$$F(x) = \begin{cases} 5 & \text{if } x = 0 \\ 4 & \text{if } x = 1 \end{cases}$$

3	if x = 2
2	if x = 3
1	if x >= 4

This translation of x, F(x), is the Actual Success Rate (ASR) and ranks in line with the RPSR, thus permitting the Confidence Ratio derivation and interpretation as described above.

Once the RPSR and ASR have been calculated, they can be inserted into equation 3.2 below and a result is obtained.

$$\text{Confidence Ratio} = \frac{Ratio\ Perceived\ Success\ Rate\ (RPSR)}{Actual\ Sucess\ Rate\ (ASR)}$$

Equation 3.2 Confidence Ratio (Thorne *et al.* 2004)

Figure 3.12 shows confidence calculations for both control and treatment groups.

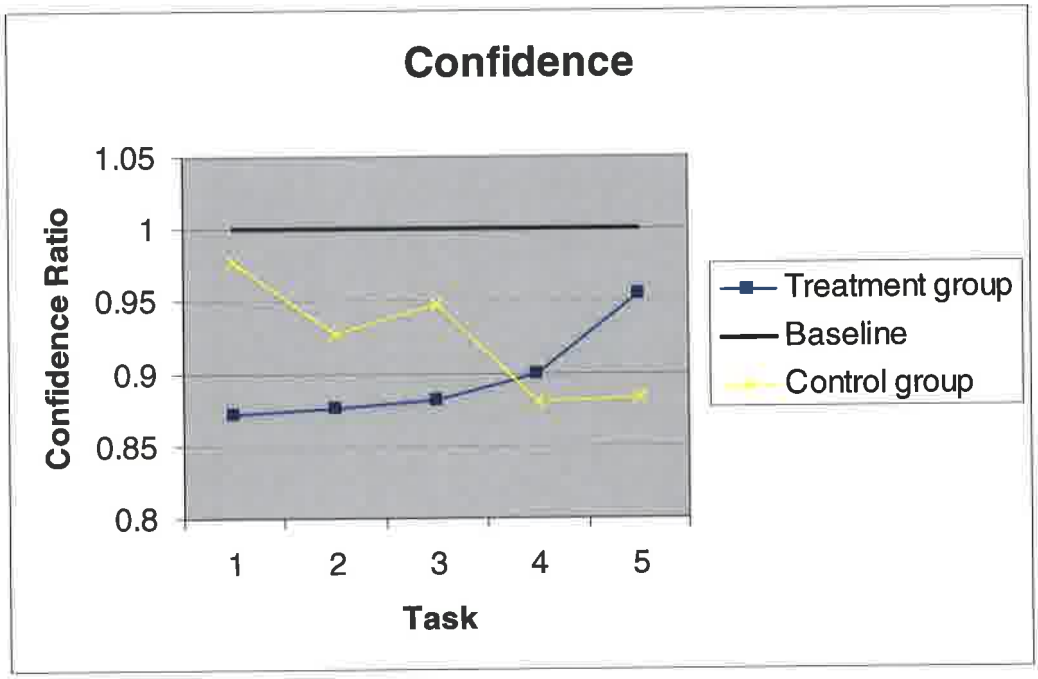

Figure 3.12 Confidence in Treatment and Control groups

The baseline on the graph shows the division between over and under confidence, a value of less than 1 indicates under confidence, over 1 indicates overconfidence. A value of 1 exactly indicates perfect calibration between expected outcome and performance.

As can be seen, both groups were under confident in their work. This is a usual finding since the literature indicates that spreadsheet developers are usually overconfident (Panko, 2003).

Although the data in figure 3.12 shows that both groups were mostly under confident, there are some distinguishing features between them.

In figure 3.13 the X axis (difficulty) and Y axis (completeness) values are defined as follows:

Value	X (Difficulty)	Y (Completeness)
1	Very Hard	No
2	Hard	Probably not
3	Average	Don't know
4	Easy	Probably
5	Very Easy	Yes

Figure 3.13 shows the responses gathered from the questions: "how difficult was this task?" (Difficulty) and "Did you complete the task successfully?" (Completeness) for both the control and treatment groups.

Please note in the figure below the directions of the axes are non standard, due to limitations in Excel.

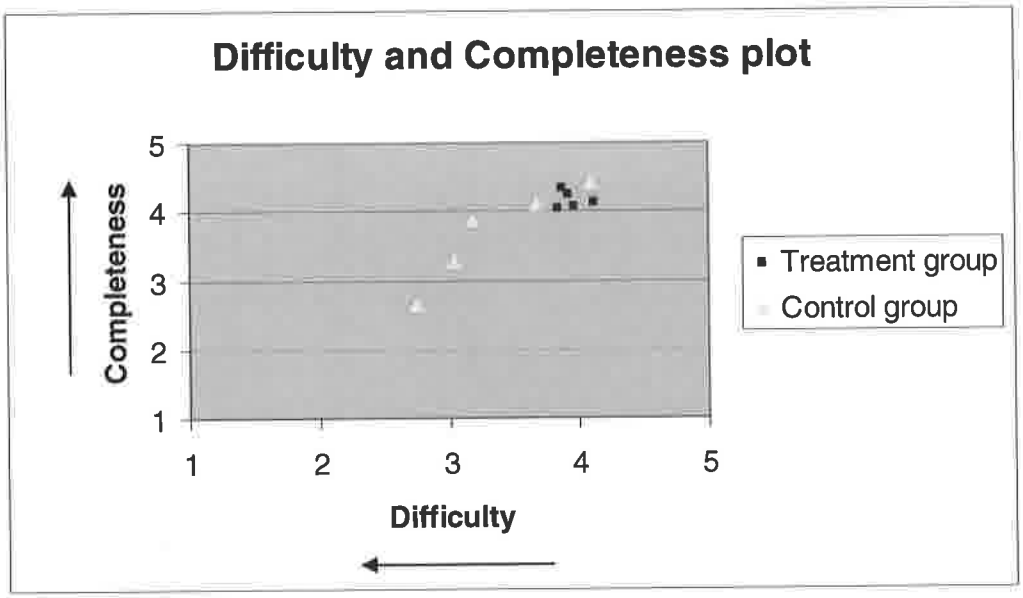

Figure 3.13 Perceived difficulty and perceived completeness scatter plot

In figure 3.13, the treatment group data points are bunched together. The values are responses to difficulty and completeness questions mapped against each other. The treatment group's data points are less erratic than the control group, indicating a more consistent approach to evaluating their performance

Figure 3.13 suggests that the treatment group found the tasks difficulty and perceived completeness didn't change as the tasks progressed. In figure 3.13, the data points read right to left as tasks 1 to 5 for each respective group

The control groups data points are more dispersed, indicating that the values change as the tasks progress, i.e. as the tasks progressed they were harder and perceived to be less complete.

3.6.4 Perceived difficulty

Perceived difficulty indicates how difficult the participants thought each question was, see figure 3.14. The data is based upon the question "How difficult was task x?" The Y axis scale relates to the level of difficulty indicated by the participant, where:

Value	Y (Difficulty)
1	Very Hard
2	Hard
3	Average
4	Easy
5	Very Easy

Please note in the figure below the directions of the axes are non standard, due to limitations in Excel.

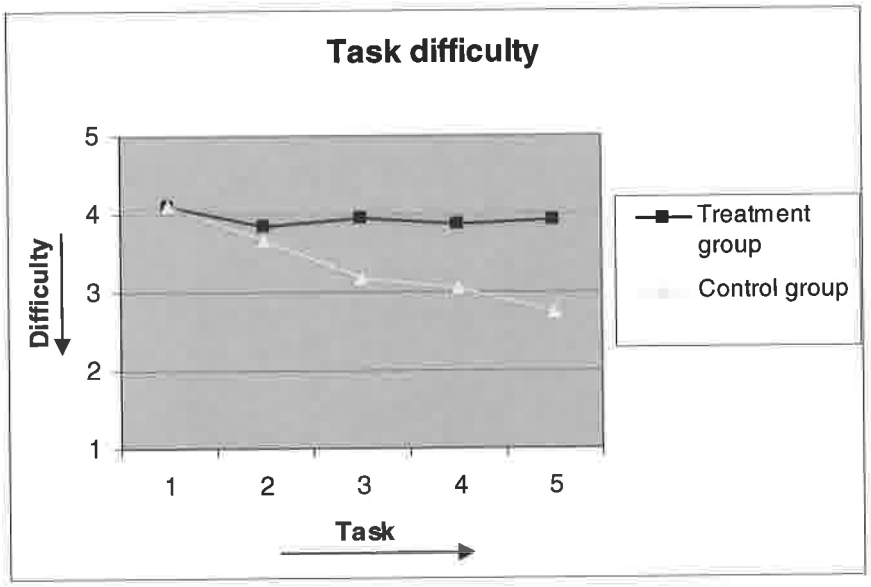

Figure 3.14 Perceived difficulty for Treatment and Control groups

Figure 3.14 shows perceived difficulty for the treatment group average around the easy classification and for the control group a range between Easy and Average. Therefore the treatment group found the tasks easier overall, i.e. giving examples is easier than constructing formulae.

This concurs with figure 3.13 which shows that perceived difficulty for the treatment group is approximately the same for all tasks. So therefore the treatment group found giving examples easier overall and giving examples is generally easier than producing formulae.

3.6.5 Perceived completeness

Figure 3.15 shows data collected from the question "Did you successfully complete task *x*?"

Primarily this is used in the confidence calculation but is interesting since the figure shows that the control group were less confident in the answers they gave than the treatment group, see figure 3.15.

The Y axis scale relates to the level of completeness indicated by the participant, where:

Value	Y (Completeness)
1	No
2	Probably not
3	Don't know
4	Probably
5	Yes

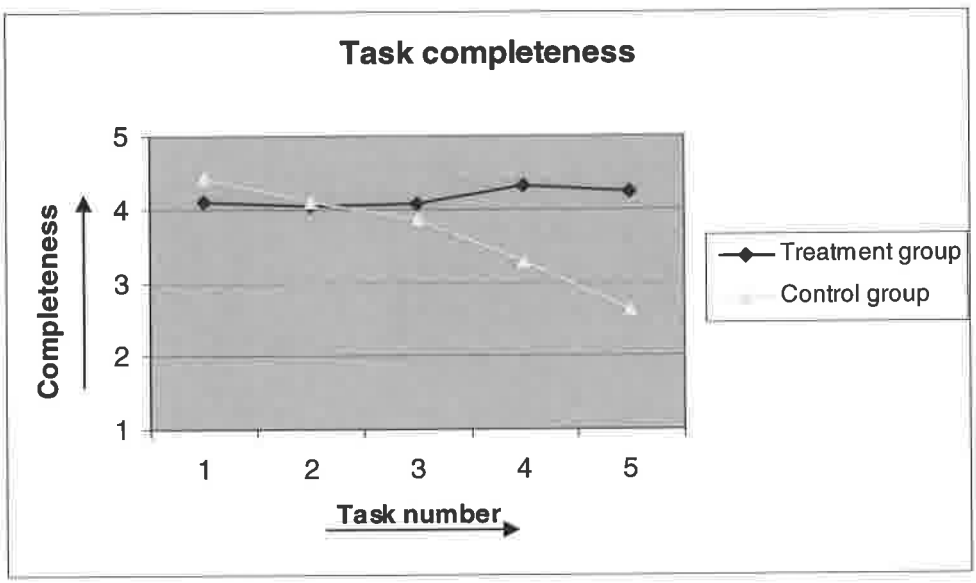

Figure 3.15 Perceived completeness

3.6.6 Conclusions on summary statistics

Comparing the accuracy between the treatment and control groups shows that in all 5 tasks the treatment group (example giving) were more accurate, see figure 3.4.

The profile of the participants in the control and treatment groups were approximately the same, see figures 3.5, 3.6 and 3.7.

The results contained in figures 3.8, 3.9, 3.10 and 3.11 shed little light on whether experience affects accuracy. This is because some of the participants have given contradictory answers.

For example when one participant was asked how experienced they were with spreadsheets they stated "competent". However, when the same participant was asked how many years they had been using spreadsheets they stated "never used them before". It is possible that some participants misunderstood the experience questions and hence any results gained from the experience questions will be disregarded.

The results from the confidence calculation, see figure 3.12, show that both the treatment and control group were under-confident. This is an unusual result since the literature suggests most spreadsheet developers are overconfident (Panko, 2003).

One possible explanation for this under-confidence is the experimental process by which they were tested. For example, the act of asking participants how confident they were in their work altered the manner in which they responded this is an example of the "Hawthorne effect" which is examined in more detail in the limitations section.

Figure 3.12 shows the treatment group participants found the difficulty of the tasks and the ability to complete the tasks more consistent than the control group participants.

This is further reflected in figures 3.13 and 3.14 which show that treatment group participants consistently found the tasks easier, figure 3.13, and felt they were able to complete more of the tasks than the control group, see figure 3.14.

The raw data for both experiments, when graphed, allows conclusions to be drawn based up some basic statistics such as the mean value. Whilst this serves a purpose, it does not tell us if the results are statistically significant.

Therefore the next section applies a number of significance tests to evaluate if the relationships present in the results are statistically significant.

Further details on the above experimentation are contained in Thorne *et al.* (2007).

3.7 Testing for statistical significance in the results

3.7.1 Introduction

In order to see if the results are statistically significant a number of significance tests were applied to the accuracy data. For example, the Chi squared test is used to determine if the differences in accuracy are statistically significant in the control and treatment groups. One can then determine if the increased accuracy observed in the treatment group was due to the treatment or not.

3.7.2 Chi-Squared (χ^2) Test

The chi-squared test measures the odds that the relationship observed in a sample of data, taken from a population, is representative of the relationship expected in the total population. In other words, the chi-squared statistic (displayed below in equation 3.3) tests whether the observed results significantly differ from the expected results if by chance.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Equation 3.3 Chi squared

Where

O_i = the observed result for outcome i, i = 1, ..., n
E_i = the expected result for outcome i, i = 1, ..., n, derived from the null hypothesis
n = number of outcomes in sample

The chi-squared test statistic, as in equation 3.3, can be used in statistical hypothesis testing to consider the null hypothesis (H_0). 'Null' means nothing; hence the null hypothesis suggests that there is nothing going on, i.e. no significant difference between observed and expected observations. The alternative hypothesis (H_1) is therefore the suggestion that there is a significant difference between the populations.

When comparing the χ^2 statistic (eq.3.3 above) with the χ^2 distribution with (n-1) degrees of freedom, we retrieve a p-value. The p-value is the probability of obtaining a test statistic greater than or equal to the value calculated in equation 3.3, given that H_0 is true. A standard commonly used significance level for testing hypotheses is 0.05. That is to say, that if $p > 0.05$ then the null hypothesis is accepted i.e. there is no significant difference between the observed and expected populations.

This basic theory can be extended to take many different forms subject to specific data constraints and the objectives of the analysis. Within the scope of this work, testing the significant difference between example-giving and control by task the data is accumulated in the form of a 2 x 2 contingency table.

Owing to the fact that different participants undertook the two distinct methodologies (treatment and control) as well as the fact that the treatment and control groups had an unequal number of participants, the following chi squared equation has been employed:

Participants	**Success**	**Failure**	**Totals**
Treatment	a	b	a + b
Control	c	d	c + d
Totals	a + c	b + d	a + b + c + d = N

Table 3.2 chi squared 2 x 2 contingency table

3.7.2.6 Summary of Chi squared

Table 3.3 contains summary information for the Chi squared tests performed on tasks 1 to 5. Full results and calculations can be found in Appendix A.

Task	Chi squared test statistic	Exact *P* value	H_0 Outcome
1	1.396	76.3%	Accept
2	0.673	58.8%	Accept
3	2.032	84.6%	Accept
4	2.032	84.6%	Accept
5	4.22	96.0%	Reject

Table 3.3 Chi squared values summary

The results show that, for a 95% significance level, the only statistically significant difference is in task 5. It is possible that the sample sizes are too small to perform the chi squared statistic on. In order to be certain another significance test was applied to the results to see if a similar situation arose.

In cases where the sample size is small, Fisher's Exact test can be used to compliment or replace the chi squared test (Fisher, 1922).

3.7.3 Fishers Exact Test

The fisher exact test can be used in place of the chi-squared test, which is not suitable if sample sizes are small. The test is, as the name suggests, exact, and it can therefore be used regardless of sample size restraints. Note that its complex calculation for large samples makes it less desirable, in this circumstance however, the chi-square test is appropriate.

The fisher exact test is used to calculate the probability of significant association between two variables based on a 2 x 2 contingency table. If an event, A, can have two possible options (A1 and A2 say) and a population, B, is split into two sub-

populations (B1 and B2 say) then we can display the exhaustive list of scenarios in the following 2x2 contingency table:

	B1	B2	Totals
A1	a	b	a + b
A2	c	d	c + d
Totals	a + c	b + d	n

Table 3.4 Fisher's exact 2x2 contingency table

Where

a = the number of cases of event A1, in sub-population B1
b = the number of cases of event A1, in sub-population B2
c = the number of cases of event A2, in sub-population B1
d = the number of cases of event A2, in sub-population B2
n = total number of cases

The probability (p-value) of obtaining such a situation is given by the hypergeometric distribution (Fisher, 1922), the formula for which is displayed in Equation 3.4 below and uses the notation detailed in Table 3.4 above.

$$\frac{(a+b)!\,(c+d)!\,(a+c)!\,(b+d)!}{n!\,a!\,b!\,c!\,d!}$$

Equation 3.4 Fisher's exact

3.7.3.1 Fisher's exact summary

Table 3.5 contains the summary information for tasks 1 to 5 obtained using Fisher's exact. The full Fisher's exact results and calculations are contained in appendix A.

Task	Fisher's exact	Probability	H_0
1	0.205	80%	Accept
2	0.301	70%	Accept
3	0.128	88%	Accept
4	0.128	88%	Accept
5	0.038	96%	Reject

Table 3.5 Fisher's exact summary

As can be seen from table 3.3, differences in tasks 1 to 4 show no statistical significance. However, task 5 does show statistical significance, this supports the chi squared statistic which shows similar outcomes, see table 3.2.

3.7.4 Summary of statistics

The combined results obtained from chi squared and Fisher's exact are contained in table 3.6.

	Chi squared (exact values)		Fisher's exact	
	P	H_0 outcome	*P*	H_0 outcome
Task 1	76.3%	Accept Null	79.5%	Accept Null
Task 2	58.8%	Accept Null	69.9%	Accept Null
Task 3	84.6%	Accept Null	87.2%	Accept Null
Task 4	84.6%	Accept Null	87.2%	Accept Null
Task 5	96.0%	Reject Null	96.2%	Reject Null

Table 3.6 Combined Chi squared and Fisher's exact statistics

The data in table 3.6 and the data graphed in figure 3.16, show that for both Chi squared and Fisher's exact, tasks 1 to 4 are not statistically significant, assuming that 95% is the minimum level of significance.

However, both show on task 5 statistical significance which therefore rejects the null hypothesis on that test. We can conclude that for task 5 the observed difference in accuracy was due to the treatment not chance.

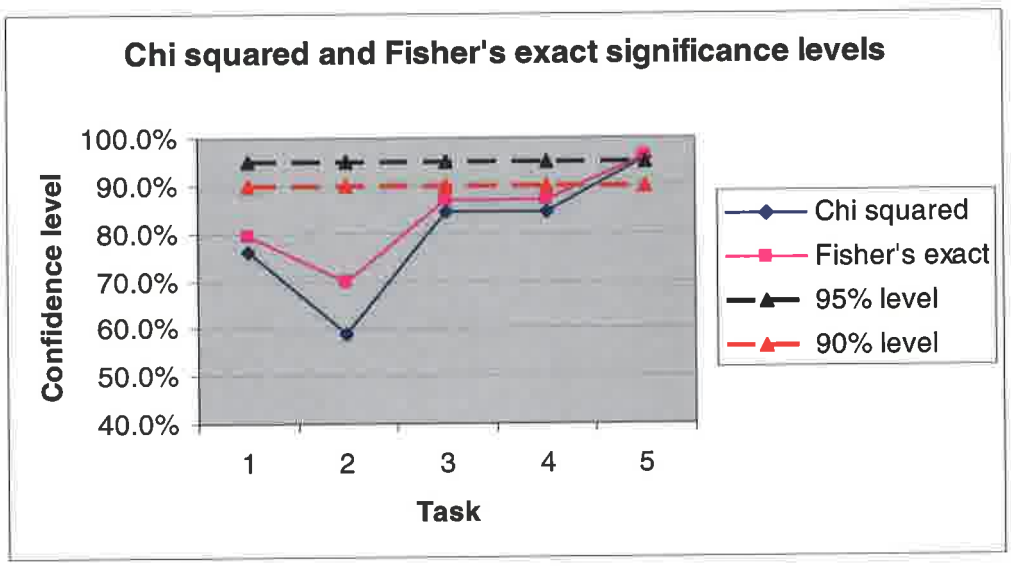

Figure 3.16 Chi squared and Fisher's exact significance levels

Since the tasks were designed to be progressively more difficult, one could interpret the results to show that the treatment is only effective in sufficiently complex scenarios.

Using Cochran's Q test determines if the difficulty between tasks was statistically significant

3.7.5 Cochran's Q Test

Cochran's Q Test is an extension of McNemar's Test, looking at comparing significant marginal frequency differences across more than 2 dimensions. That is, a b x k contingency table, where b and/or k can be > 2. In the context of the EDM experiment, this allows us to test for any significant difference in results between the five tasks.

The Cochran Test statistic is as depicted in Equation 3.5 below:

$$Q = \frac{k(k-1)\sum_{j=1}^{k}\left(x_{.j} - \frac{N}{k}\right)^2}{\sum_{i=1}^{b} x_{i.}(k - x_{i.})}$$

Equation 3.5 Cochran's Q (Cochran 1950)

Where

k is the number of treatments (or tasks)
$X_{\bullet j}$ is the column total for the jth treatment (or task)
b is the number of blocks (or results)
$X_i \bullet$ is the row total for the i^{th} block (or result)
N is the grand total

For the Cochran Q Test, the critical region, for significance level α (0.05 for instance), is $Q > {}^{\chi}{}_{1-\alpha,k-1}{}^2$ where $\chi_{1-\alpha,k-1}$ is the $(1-\alpha)$ quantile of the χ^2 distribution with (k-1) degrees of freedom. That is to say that if this condition is met, i.e. Q falls in the critical region, the null hypothesis is rejected inferring that the performance results for the 5 tasks differ significantly. If all tasks are conducted with controlled static

conditions, then this significant difference in results can furthermore be attributed to a change in difficulty level across the tasks.

3.7.5.1 Cochran's Q for the Control group

The calculation for Cochran's Q statistic in the control group is as follows:

$5 * 4 * (16 + 4 + 1 + 1 + 16)$
$= 760 / (270 - 194)$
$=10.00$

DOF = 4
$0.05 < P < 0.02$

This shows that there is a significant difference in difficulty between tasks for the control group, we reject the null hypothesis at the 95% level.

3.7.5.2 Cochran's Q for the Treatment group

The calculation for Cochran's Q statistic for the treatment group is as follows:

$5 * 4 * (10.24 + 0.04 + 0.64 + 0.64 + 3.24)$
$=296/(390-364)$
$=11.386$

DOF = 4
Look up on Chi Squared table
$0.05 < P < 0.02$

This shows that there is a significant difference in difficulty between tasks for the treatment group, we reject the null hypothesis at the 95% level.

3.7.5.3 Conclusions on Cochran's Q test

The calculations of Cochran's Q test show that at the 95% confidence level, the null hypothesis is rejected for both control and treatment groups. This implies that there is a significant difference between tasks, see section 3.7.5.2 and that this difference is attributed to increasing difficulty, see last sentence in section 3.7.5.

However, tasks 3 and 4 both show the same result for chi squared and Fisher's exact, see table 3.6. This might suggest that these two tasks were of similar difficulty based on the results.

In order to establish if this is the case, we must compare the two sets of data for the control and treatment group to see if there is statistical significance between them. One method to compare two data sets for difference in difficulty is McNemar's test on difficulty (McNemar, 1947).

3.7.6 McNemar's Test

McNemar's test statistically compares two proportions for dependence or correlation. It does this by determining whether row and column marginal frequencies are statistically similar. Its application is for 2x2 contingency tables (see figure 3.7), as for the Fisher Exact Test, and can be employed to test the marginal homogeneity for results A and B, when tasks x and y are performed.

	Task x		
Task y	**A**	**B**	**Totals**
A	A	B	a + b
B	C	D	c + d
Totals	a + c	b + d	n

Table 3.7 McNemars test 2 x 2 contingency table

Where

a = the number of cases of result A, in tasks x *and* y
b = the number of cases of result B in task x, and result A in task y
c = the number of cases of result A in task x, and result B in task y

d = the number of cases of result B, in tasks x *and* y
n = total number of cases

In the context of the EDM experiment, this allows us to test 2 results (A and B), to investigate whether the performance for tasks x and y are significantly similar or not. If the conclusion is of significant difference between marginal frequencies, then the follow-on assumption, given all other conditions remain equal, could be of a difference in difficulty of the task.

The McNemar Test Statistic shown below, in Equation 3.6 is based on the concept of marginal homogeneity and looks at row and column frequencies being the same, as follows: for result A, a + b = b + c, and for result B, c + d = b + d. These two equations then reduce to test whether b = c, as demonstrated in the test statistic.

$$\chi^2 = \frac{(b-c)^2}{b+c}$$

Equation 3.6 McNemars test

This is a chi-squared test statistic and has 1 degree of freedom. The significance testing on this statistic is the same as for the χ^2 statistic; the p-value is derived from the chi-squared distribution and then tested for significance. If $p > 0.05$ then the null hypothesis is accepted, this means the performance results, A and B, are significantly similar across tasks x and y.

The McNemar's statistic allows us to test for significant difference in difficulty between the two groups, in this case the results for task 3 and 4, see tables 3.8 and 3.9.

The test is X^2 using 1 DOF.

	Fail	Pass
Fail	10	3
Pass	3	7

Table 3.8 McNemar's test, Control group

$M = (3-3)^2 / (3+3) = 0/6 = 0$

We therefore accept the null hypothesis, there is no difference between the two groups i.e. there is no significant difference in difficulty between tasks 3 and 4 for the control group.

	Fail	Pass
Fail	7	2
Pass	2	14

Table 3.9 McNemar's test, Treatment group

$M = (2-2)^2 / (2+2) = 0/4 = 0$

We therefore accept the null hypothesis, there is no difference between the two groups, i.e. there is no significant difference in difficulty between tasks 3 and 4 for the treatment group.

3.7.7 Conclusions on significance testing

The chi squared and Fisher's tests indicate that in both the control and treatment groups, for tasks 1 to 4, there is no statistically significant difference in accuracy.

However, both chi squared and Fisher's indicate that for task 5, in both control and treatment groups ,the observed increase in accuracy is statistically significant. i.e. the difference in accuracy is due to the treatment and not chance, ergo giving examples in task 5 is more accurate than producing the equivalent formula. See table 3.6 and figure 3.16 for a summary of all the results.

Cochran's Q test indicates that between all five tasks, there is a significant difference in difficulty. McNemar's test on the observed accuracy in tasks 3 and 4, which have the same values, demonstrates that there is no significant difference in difficulty between the tasks.

One possible explanation is that during the design of the materials, i.e. the tasks were not sufficiently different to yield a significant change in difficulty, hence the same accuracy values.

To conclude, there is a relationship between difficulty and statistically significant accuracy for the treatment. The results suggest that if the task or problem is sufficiently difficult, there is a statistically significant accuracy advantage in using the treatment over the control, **i.e. the treatment effect is significant as complexity increases.**

3.8 Conclusions on feasibility experiment

The conclusions of the experimental comparison between the Treatment group, i.e. giving examples and control group, i.e. producing formulae

3.8.1 Experimental Conclusions

1. The treatment group (giving examples) were considerably more accurate than the control group (producing formulae), see figure 3.6. Accuracy in task 5 was the only task to be statistically significant, see table 3.6 and figure 3.16.

2. Both the treatment group (giving examples) and the control group (producing formulae) were consistently under confident, see figure 3.12.
3. The treatment group consistently found the tasks easier than the control group, see figure 3.14. Further, both groups found the tasks progressively more difficult as Cochran's Q test indicated, except tasks 3 and 4 which showed no significance of this type, see section 3.7.5
4. **The treatment effect is significant as complexity increases, see section 3.7.7.**

3.8.2 Limitations

Limitations to this experimental study include both general criticisms of experimental work and specific conditions that relate to the experiment. Also criticism could be made of the statistical significance tests due to the way that they are marked.

3.8.2.1 The fair test of the novel approach

The most significant criticism of this experiment is the apparent comparison of two seemingly different tasks, one is giving-examples the other is programming a spreadsheet. Ideally the example-giving group would have a complete working system rather than just example-giving on paper, i.e. they would give examples and these examples would then be converted into a working model. This would have been a fairer more comparative test of the example-giving method and spreadsheet development.

However, because this work was conducted at an early stage of the research and the emphasis of the experiment was placed on how feasible the technique of example-giving was the decision was made to proceed as described in this chapter.

Moreover, the mechanics of converting examples into working models is explored in depth in later chapters. The most critical aspect of the example-giving method is the actual production of valid examples which this experiment investigates in detail. If valid examples for a particular problem can be generated, the process of converting them into a model via whatever means is less critical than production of valid

examples. If the experiment proved it was infeasible to use example-giving, the actual means of implementing it becomes irrelevant.

3.8.2.2 Bias present in the sample of participants

Examining the responses on spreadsheet experience gathered from the participants, figures 3.5, 3.6 and 3.7, some inconsistency between groups is evident. In particular, it appeared that the treatment group contained participants with a significant experience bias in comparison to the control group, as shown in figure 3.6.

For this reason a one-way Analysis of Variance (ANOVA) was performed on the responses contained in figure 3.6. The procedure and detail of one-way ANOVA is discussed at length in Plonsky (2006). As a high level overview, analysis of variance compares group means by analysing comparisons of variance estimates.

A one-way ANOVA compares two (or more) populations for significant variation in their mean value. In this case ANOVA was used to test the distribution of responses pertaining to spreadsheet experience in years. That is to say, are treatment and control groups significantly different in terms of participant spreadsheet experience?

The first step within ANOVA is to calculate the between group sum of squares (SS_{bg}) and the within group sum of squares (SS_{wg}), as follows:

$$SS_{bg} = n\sum_{j}\left(\overline{Y_j} - GM\right)^2 \qquad SS_{wg} = \sum_{i}\sum_{j}\left(Y_{ij} - \overline{Y_{ij}}\right)^2$$

Equation 3.7 ANOVA statistic

Where,

GM = the grand mean over all N observations,

$\overline{Y_j}$ = the sample mean for each group j,

Y_{ij} = the frequency of participants in experience level i, group j,

$\overline{Y_{ij}}$ = the samples mean for each frequency i, group j,

n = the number of observations in each group.

Once SS_{bg} and SS_{wg} are calculated, the ANOVA table can be set up as shown here:

ANOVA				
Source of Variation	*SS*	*df*	*MS*	*F*
Between Groups	SS_{bg}	j-1	MS_{bg} = SS_{bg} / j-1	F = MS_{bg} / MS_{wg}
Within Groups	SS_{wg}	N-j	MS_{wg} = SS_{wg} / N-j	
Total	SS = SS_{bg} + SS_{wg}	N-1		

For this scenario, the results of the one-way ANOVA are shown in Table 3.10 below.

ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	2.282645	1	2.282645	2.323225	0.134302	4.051749
Within Groups	45.19652	46	0.982533			
Total	47.47917	47				

Table 3.10 ANOVA results

In table 3.10 the null hypothesis (the groups are not significantly different) is rejected if the *F* value is higher than the *F crit* value. As can be seen here, the *F* is lower than the *F crit* (with a critical value of 0.05) and thus we accept the null hypothesis. Hence, the groups are not significantly different, i.e. no significant experience bias is present between the treatment and control samples.

However, it is still worthy of note that the sample of participants was from an academic environment, experimentation with participants from a non academic environment would provide a broader view of the usefulness of this method.

3.8.2.3 Experimental conditions and the Hawthorne effect

Although there was no time limit imposed on the participants to complete the tasks, participants were not permitted to take the materials away from the venue. Some

might argue that this imposes a time pressure on the participants and that in reality they are more likely to complete the tasks over a longer time period.

However, to keep control of the experimental conditions one must insist that participants stay in the arranged venue until they have completed. Allowing them to remove and complete materials at another venue may allow collusion and thus the integrity of the experiment would be compromised.

One must consider the "Hawthorne effect" which is often cited as a flaw in experimental methods. Parsons (1974) defines the Hawthorne effect as follows:

> *"I would define the Hawthorne Effect as the confounding that occurs if experimenters fail to realize how the consequences of subjects' performance affect what subjects do"*

So in the context of this experiment, the fact that the participants are asked to do something changes the way in which they do it. This in turn leads to results that are not truly representative of the naturally occurring phenomena, i.e. the work produced for the experiment is not a true representation of their ability.

By definition, the Hawthorne effect is impossible to avoid but to ensure the objectivity and representative nature of experimental work, rigorous methodologies and designs are followed. This experiment was designed using two seminal experimental design texts, Campbell and Stanley (1969) and Shadish *et al.* (2002). The use of such materials mitigates the Hawthorne effect as much as possible.

Central to rigorous design is the concept of truly random samples. It could be argued that the sampling approach taken in this experiment is not truly random. A clustered random approach was taken, i.e. a cluster of individuals were targeted and then randomly assigned to either the treatment or control group.

Once the cluster was identified, participant selection was randomly assigned within the cluster, i.e. the participants were not selected on ability or age or gender or any

other basis. Thus the control and treatment groups were not biased, i.e. the treatment group participants didn't have greater academic ability.

Although the study has limitations, it still provides a controlled measure of the usefulness and potential accuracy of an example based approach to spreadsheet modelling.

3.8.2.4 Criticisms of the significance testing

The significance tests show that only task 5 is statistically significant. The Cochran's Q statistic shows that the difficulty difference between the tasks is statistically significant.

The tasks were designed to be progressively more difficult. The conclusion is therefore that the treatment effect is only statistically significant in sufficiently difficult tasks.

However, as can be seen in figure 3.16 and table 3.6, the chi squared and Fisher's values show an irregular relationship between probability and task. One would expect to see the values rising progressively as the tasks progress if the conclusion drawn was true.

Despite this, tasks 1 to 4 are still less difficult than task 5, so the conclusion that the treatment is only statistically significant in sufficiently difficult tasks still holds true, what remains unclear is the relationship between significant accuracy and tasks 1 to 4.

The statistics generated from the raw data are sensitive to the marking applied to the answers provided to each question. The answers were dichotomous, i.e. attempts were either correct or incorrect. In both the control and treatment group this mark was based upon whether the solution provided was a valid solution that covered the specification of the task.

If the method used to mark the answers provided for each task differed, one would expect to see a change in the statistics. If the statistics were calculated data that had been processed according to an invented marking criteria, the sensitivity of the statistics would be greater.

However, since all of the statistics were strictly marked in a dichotomous fashion, this sensitivity is not a limiting factor in this research.

3.9 Advantages and disadvantages of the example-giving approach

This section seeks to summarise the main advantages and disadvantages gained from using an example-giving approach.

Advantages of example-giving:

- Example-giving is easy (see section 3.6.4)
- Eliminates the need to program the computer
- Eliminates BER in the programming of a spreadsheet
- Eliminates bias in the spreadsheet

From the above advantages of example-giving, it would seem to be that many sources of the problem of spreadsheet error (section 2.7) such as BER, Poor programming and bias are reduced or eliminated.

Disadvantages of example-giving:

- May introduce some BER in the proposed alternative method, i.e. creation of examples
- May introduce bias in the creation of examples

Further questions and detail regarding example-giving are addressed in later chapters.

3.10 Summary on feasibility of example-giving

The conclusions of the feasibility experiment are contained in sections 3.5.6, 3.6.6 and 3.7.7 dealing with feasibility experiment design, conclusions on summary statistics and conclusions on significance testing respectively.

Section 3.5.6 concludes the design details that give the experiment the objectivity to be a fair test between example-giving and traditional spreadsheet modelling.

Section 3.6.6 concludes that the summary statistics show the treatment group (example giving) were more accurate, found the tasks easier and felt they were able to complete more of the tasks than the control group (traditional spreadsheet modelling). The summary statistics show that both the control and treatment groups were under confident, i.e. both groups performed better than they anticipated. However, judgement of success was better in the treatment group than the control group.

Section 3.7.7 concludes that the superior accuracy shown by example-giving is only statistically significant in the last task, task 5. The use of McNemar's test on difficulty proves that the tasks were progressively complex, excluding tasks 3 and 4. This suggests example-giving is more useful in more complex scenarios.

The above conclusions show that example giving is a feasible novel approach for decision support spreadsheet modelling (see section 3.1.1) that can offer a statistically significant accuracy advantage to traditional methods.

These findings partially satisfy objective 2:

> *"Based upon the literature review, consider an alternative modelling technique for the reduction of error in decision support spreadsheets*

This chapter proves that the alternative paradigm of example-giving is feasible and can work in principle. However it does not consider how example-giving might be implemented, this issue is dealt with in the next chapter.

From these experiments comparing example-giving with conventional spreadsheet construction would suggest some further work would be useful on the following points:

1. Example giving appears to be much faster (practical observation of the groups).
2. To what extent this would scale up to a very large scale requires further investigation.
3. Example-giving is perhaps not a familiar approach, in practice the user may not be confident in using it.

4.0 Designing the implementation of example-giving

4.1 Overview of the chapter

Section 4.2 introduces the rational for the content of this chapter. Section 4.3 examines and analyses the potential approaches that could be taken to implement example-giving. Section 4.4 considers techniques available in the selected approach, machine learning in more detail, settling on one machine learning approach. Section 4.5 discusses how experimentation could be conducted using neural networks and considers what neural network software is appropriate for experimentation. Section 4.6 discusses in detail the design of the neural networks to be used in experimentation. Section 4.7 offers a compact summary of the conclusions drawn from sections 4.5 and 4.6 which give the design of the neural networks to be used in experimentation. Section 4.8 summarises the main benefits gained from using neural networks as a means to implement example-giving. Finally section 4.9 concludes the work of the chapter.

4.2 Introduction

Since the feasibility experiment in the previous chapter shows that example giving is feasible, one must decide upon an approach to implement EDM.

The process of transforming examples into a model that represents those examples can be achieved a number of different ways.

Examples of appropriate methods are narrowed to those that aid decision making either by classification or decision mapping.

4.3 Approaches to implementing example-giving

The appropriate methods considered are: Karnaugh maps; The Quine–McCluskey algorithm; The Espresso heuristic logic minimiser, Decision trees and machine learning.

4.3.1 Karnaugh maps

Karnaugh maps (Karnaugh, 1953) are a minimisation technique for Boolean functions, in other words Karnaugh maps simplify logic based equations.

The minimisation assumes variables as binary values i.e. each variable has two states (1 or 0). These binary states are used to create a truth table which is then converted into a grid. See figure 4.1

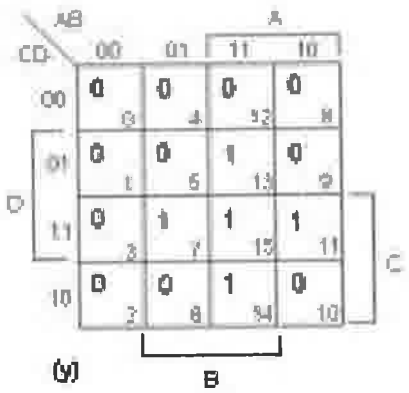

Figure 4.1 Four variable Karnaugh map

Once in grid form, variable combinations in the grid can be ignored or combined to reduce the number of expressions contained in the table.

The process of minimisation relies on the human participant detecting patterns in the data and combing terms to give the simplest expression of the equation.

4.3.1.1 Advantages to Karnaugh maps

Problems with less than 4 variables can be converted easily into Karnaugh maps and allow quick minimisation of Boolean equations.

4.3.1.2 Limitations to Karnaugh maps

The major limitation to this method is the reliance on the human's ability to pattern match a large number of variables. The Karnaugh map size is determined by the equation: 2^{n-1}

Once more than 4 variables are reached, the resulting grid becomes too unwieldy for a human to minimise, i.e. it becomes impractical to use.

For example a 10 variable problem (2^{10-1}) will have a corresponding Karnaugh map will have 1023 unique cells. Clearly a human trying to evaluate that number of expressions will have difficulty with the sheer volume. This problem may be overcome with computation.

Karnaugh maps were designed to be manually created and therefore directly automating Karnaugh maps is highly inefficient. In answer to this an altered version of Karnaugh maps designed for computation were introduced in the Quine-McClusky algorithm.

4.3.2 Quine-McClusky algorithm

The Quine–McCluskey algorithm (Quine 1952, McClusky and Bartee 1962) works in the same manner as Karnaugh maps except the generation and reduction of grids is executed by computation rather than a human manually pattern matching.

4.3.2.1 Advantages of the Quine–McCluskey

The Quine–McCluskey algorithm is an improvement the usefulness of Karnaugh maps on the basis that more variables can be assessed via computer automation.

A heuristic search algorithm is used on the tables to combine variables and arrive at the simplest Boolean expression.

4.3.2.2 Limitations to the Quine–McCluskey algorithm

According to Heiber (2007) the Quine-McCluskey algorithm suffers similar practical limitations as Karnaugh maps since the size of the grid increases exponentially. The rate at which the grid size increases is $3^n/n$ where n = the number of variables in the expression.

For example if there are 20 variables the corresponding grid will have 174,339,220 unique cells.

Eventually this exponential growth causes the grids to become so large the heuristic method either cannot compute the best result or takes too much time to compute.

This long computation is mitigated by using the Espresso heuristic logic minimiser.

4.3.3 Espresso heuristic logic minimiser

The Espresso heuristic logic minimiser, developed by the University of California Berkeley, improves on Karnaugh maps and the Quine-McClusky algorithm by minimizing the resources needed to compute complex grids.

However, the espresso algorithm is only capable of achieving a *close* to optimal solution.

Combining the espresso heuristic logic minimiser with the Quine-McCluskey algorithm allows close to optimal creation of complex maps, however the basic problem of a human interpreting these maps remains.

4.3.4 Decision trees

Decision trees, which are sometimes referred to as classification trees, map the reasoning process, the dependant variable, the independent variables and the classifications of a decision process (Quinlan, 1986).

The dependant variable is the focus of the decision process, for example this could be deciding whether to play golf or not

The independent variables affect the decision process, for example the weather outlook, temperature, humidity and wind levels.

Classifications are the conclusions that are based upon the independent variables and the rules. In the golf example the classifications are either play golf or don't play golf.

The classification tree is built up from a data set of examples and the rules of the inferred from the data. See table 4.1 for a simple data set for the golf example.

Independent Variables				Dependant variable
Outlook	**Temperature**	**Humidity**	**Windy**	**Play**
Sunny	85	85	FALSE	Don't play
Sunny	80	90	TRUE	Don't play
Overcast	83	78	FALSE	Play
Rain	70	96	FALSE	Play
Rain	68	80	FALSE	Play
Rain	65	70	TRUE	Don't play
Overcast	64	65	TRUE	Play
Sunny	72	95	FALSE	Don't play
Sunny	69	70	FALSE	Play
Rain	75	80	FALSE	Play
Sunny	75	70	TRUE	Play
Overcast	72	90	TRUE	Play
Overcast	81	75	FALSE	Play
Rain	71	80	TRUE	Don't play

Table 4.1 Golf training set

Figure 4.2 shows the resulting classification tree generated from the data in table 4.1.

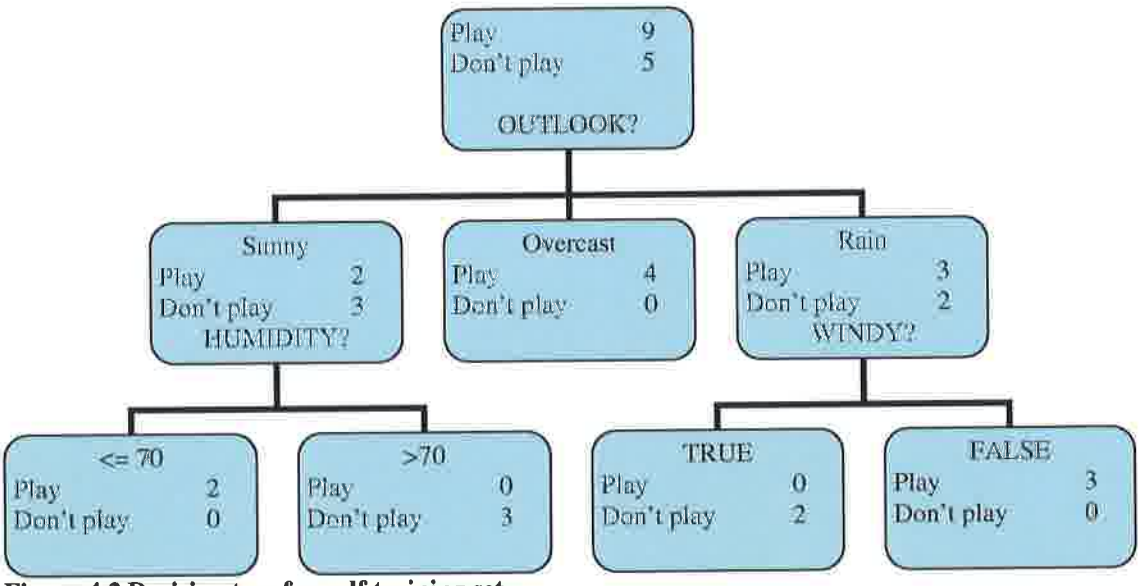

Figure 4.2 Decision tree for golf training set

As can be seen from figure 4.2, the most influential independent variable in the tree is Outlook, i.e. outlook influences the dependant variable (to play golf or not) more than the others.

4.3.4.1 Advantages to decision trees

Decision trees are easily provable because of the hierarchical structure used to generate them. By simply following the flow of a classification one can determine the rules that make that classification true. For example when the outlook is sunny and the humidity is below 70, play golf.

Further, the rules that make up classifications can easily be converted into IF THEN ELSE rules. For example consider figure 4.2 when the outlook is rain:

IF outlook = rain **AND** Windy = FALSE **THEN**

Play golf

ELSE

Don't play golf.

Because of this simplicity, small decision trees are quickly understandable by visually inspecting the path of a classification.

4.3.4.2 Limitations to decision trees

One limitation to manually creating decision trees is the amount of time it takes to create a relatively simple classification tree. This can be overcome by computer automation.

With computer automation the amount of time needed to create a decision tree is cut down but if 'numerical data sets' are being used the resulting tree can become physically large and complex.

This is because numerical values have to be interpreted as binary values, i.e. each numerical value in a range needs its own branch. So if there were a range of 10 values for one influential variable, each value needs its own branch.

Eventually the decision tree becomes too large to visualize and therefore it becomes more difficult to understand.

Further, the algorithmic process of creating a decision tree is unstable, slight changes to the training set can create entirely different decision trees. The implication of this is that if there were any user error, the error would severely impact on the decision tree.

4.3.5 Machine learning classification algorithms

Machine learning algorithms that classify data are commonly referred to as 'Classifiers'. There are several types of classifier that could be implemented; the types vary depending on the underlying algorithms used to develop classifications.

Machine learning algorithms that can classify are: Neural Networks (NN); Genetic Algorithms (GA); Inductive Logic Programming (ILP) and Inductive Expert Systems (IES).

Learning in machine learning algorithms is achieved via learning algorithms which either deductively or inductively determines the rules present in the data sets.

Some of these machine learning techniques, NN and GA, are analogies of biological systems such as animal brains and evolutionary genetics, others, IES and ILP are based on symbolic systems, like language.

4.3.5.1 Advantages to machine learning algorithms

Certain machine learning algorithms, such as IES and ILP are easily provable in much the same manner that decision trees are. In the case of IES the rules are explicitly output to a rule base, an element of IES. Further, ILP output closely resembles mathematical equations which can then be proven to be correct.

Tolerance of user error and noisy data sets is a feature of some machine learning algorithms, this is especially true of NN and GA. This allows NN and GA to learn adequately from data sets that are incomplete or contain errors and maintain a good level of performance.

Machine learning algorithms are relatively more automated than the other topics discussed in this section, i.e. there is less effort required to create a working solution.

All machine learning algorithms are computerised and can deal with varying complexity relatively well when compared to Karnaugh maps and decision tree learning.

4.3.5.2 Disadvantages to machine learning

Some machine learning algorithms such as NN and GAs are not provable, these techniques are often referred to as 'black box' applications, i.e. the mathematical process of transforming input into a working model cannot be reverse engineered.

Although NN and GA can cope well with noisy data sets, ILP and IES do not cope as well. However, ILP and IES do fair better with noise than decision trees and Karnaugh maps.

4.3.6 Test Driven Development

Test Driven Development (TDD) is an agile method for developing specifications and code for software products.

According to Ambler (2008) in TDD the developer writes test cases first and then writes code that attempts to satisfy each test. If the code satisfies the test, the developer moves on to the next test case otherwise the developer modifies the code until it satisfies the test. This process is repeated until the developer cannot think of any other test cases.

4.3.6.1 Advantages of TDD

The main advantage of TDD is the development of *higher quality* code when compared to code that is traditionally tested. George and Williams (2003) showed that TDD code passed 18% more functional black-box test cases.

TDD often results in quicker overall development and implementation times (Muller and Padberg, 2007). Also developers using TDD indicated that the process was enjoyable as George and Williams (2003) notes.

Writing test cases before code forces the developer to think about the specification and design of the software before they write it, as noted by Rust *et al.* (2006). In order to write good TDD test cases, the developer must consider the input and corresponding output of the model.

A typical test case for TDD will consist of input and expected output, the point being to write code that transforms the input into the output correctly. Consider the following example taken from Rust *et al.* (2006):

1. *A mark below 40 yields a "Fail" grade*
2. *A mark of 40 more up to and including 70 yields a "Pass" grade*
3. *A mark of 70 or more up to and including 100 yields a "Honour" grade*

Example test case data for the above could consist of:

- Mark: 26, Grade: Fail
- Mark: 41, Grade: Pass
- Mark: 75, Grade: Honour

The above TDD cases closely resemble the examples generated from the example-giving process. This suggests TDD could be a viable method for implementing example-giving.

4.3.6.2 Disadvantages of TDD

The main disadvantage to TDD is the quality of the code is entirely dependant on the quality of the tests written, which in turn is dependant on the individual's understanding of the problem at hand. If the developer writes poor tests the resulting code will also be poor.

In the context of implementing example-giving, using TDD as a means to write code based on the examples gathered during the example-giving phase would still require developers to write code. Whilst this approach is of interest and has potential for building better quality spreadsheets, it still leaves the potential for errors to arise in programming the spreadsheet.

4.3.7 Case Based Reasoning

According to Darlington (2000) and Aamodt (1994) Case Based Reasoning (CBR) is a method where problems are solved based upon previous cases with an emphasis on the reuse of solutions.

When a problem is encountered a search is performed through previous cases and solutions for a similar problem to the one at hand.

The process by which CBR is governed (Aamodt, 1994) is referred to as the four Rs:

1. *Retrieve the most similar case*
2. *Reuse the information and knowledge in that case to solve the problem*
3. *Revise the proposed solution*
4. *Retain the parts of the experience that are likely to be used for future problem solving*

CBR is similar to some machine learning approaches such as Expert Systems (ES) and Knowledge Based Systems (KBS) (Luger 2005). Both ES and KBS use a database of stored facts and through interaction with the user interrogate this

knowledge to perform a task such as diagnosis or classification. The critical difference is that ES and KBS use inference and chaining to create rules that operate in the problem domain, whereas CBR uses heuristics.

CBR has the ability to *learn* new cases and thus increase the base of knowledge on which further cases are solved.

CBR as a means of implementing example-giving could work as follows: Firstly examples are gathered and stored as cases, for example:

Case 1 Mark: 61, Grade: Pass
Case 2 Mark: 32, Grade: Fail
Case 3 Mark: 77, Grade: Honour

So when a new case is encountered, Mark in the above example, a similar past case is sought to provide the answer.

4.3.7.1 Advantages to CBR

Luger (2005) defines the main advantages to CBR as follows:

Typically in CBR cases are encoded directly and are not transformed into hierarchical knowledge as in ES or KBS, i.e. the knowledge in cases is stored simply and clearly.

In comparison to rule based methods, solving a new problem is significantly quicker since search for a similar case is quicker than generating a new rule.

Extensive knowledge of the domain is not required. In rule based systems the rules must be extracted in order for the system to function, this requires a deep understanding of the problem domain. In CBR the "rules" are additional, i.e. judgement is passed on an expanding number of cases, therefore a shallow knowledge of the domain is adequate for CBR to used effectively.

Further as Darlington (2000) notes, the process of retrieving previous similar cases and adapting them to suit a new problem is similar to way in which humans learn. This similarity allows good natural and instinctive interaction between human and computer.

4.3.7.2 Disadvantages to CBR

CBR has three major disadvantages as outlined by Luger (2005):

Cases do not include deeper domain knowledge, i.e. the shallow knowledge contained in cases may lead to inappropriate use of knowledge.

In large case bases, there are storage/compute tradeoffs, i.e. large case bases can become unwieldy and costly to store.

It is difficult to determine good criteria for indexing and matching cases. Currently retrieval vocabularies and matching algorithms are individually crafted. This obviously takes time and effort and nullifies some of the advantages of CBR.

As a means of implementing example-giving the indexing and matching issue is of significant concern. In the example in section 4.3.7, it is clear that all cases are similar so distinguishing which is the *best* will be problematic.

Possibly the most significant difficulty is the inability for CBR to generalise effectively. The example in section 4.3.7 would require many examples for a reliable case base to be built, i.e. a significantly higher proportion of examples would be needed for CBR than some other approaches.

4.3.8 Conclusions on approaches to implementing example-giving

The chosen approach to implement example-giving is machine learning, this is because in comparison to the other potential approaches it is more automated, is

superior at coping with complexity and noisy data sets and is offers a more stable repeatable process than other approaches.

Karnaugh maps, the Quine-McCluskey algorithm and the Espresso heuristic logic minimiser, which are all essentially based upon the same idea, are dismissed because of the lack of natural ability for these methods to deal with complexity and to express the results in a way which will be easily understandable.

Test Driven Development is dismissed since it still requires the programming of the computer. However, TDD does share strong similarities with example-giving and may provide interesting further opportunities for future research by combining TDD research with the example-giving research.

Case Based Reasoning is dismissed for several reasons, firstly the lack of a satisfactory indexing and searching method for previous examples would be problematic. The difficulty in generalizing to unseen examples is also a weakness, in comparison with other methods, CBR would need to gather a substantially larger number of examples to work effectively.

Decision trees, although an interesting approach to classifying data, are dismissed because of their algorithmic instability and sensitivity to noisy data sets.

4.4 Discussion of available machine learning algorithms

What follows is a critical review of the various machine learning algorithms that could be used to implement example- giving.

4.4.1 Inductive Expert Systems (IES)

There is a lack of literature concerning Inductive Expert Systems (IES) mostly due to its eventual evolution into Inductive Logic Programming. However, some literature exists on the benefits and limitations of the technology.

IES generate rules from properly formatted historical data. Expert systems, i.e. not inductive, use predefined rules to classify data sets Gross (1988).

In addition IES do not use built in logic and reasoning as conventional expert system do. The IES rule base is built on similarities of input output features to generalise to the unseen population.

Early IES utilised the ID3 (Iterative Dichotomiser 3) (Quinlan, 1986) algorithm to generates decision trees from historical data sets.

Quinlan (1990) later introduced FOIL (First Order Inductive Learner) which was perceived to be a greater success which allowed for the first time practical induction of relational rules (Russell and Norvig, 2003)

Muggleton and Buntine (1988) introduced CIGOL (which is LOGIC spelt backwards) which was seen as the beginnings of Inductive Logic Programming.

Although there are still some publications using the term "inductive expert systems" such as Mookerjee (2001) the term is much less common, ILP seems to have superseded IES.

4.4.2 Inductive Logic Programming (ILP)

Inductive logic programming is based on first order predicate calculus, the term was coined in 1991 by S. Muggleton (Muggleton, 1991).

ILP typically uses the CIGOL (Muggleton and Buntine 1988) and later PROGOL (Muggleton 1995, Muggleton 1997).

In ILP, predicate descriptions (features) are formed by taking example input, both positive and negative in combination with background knowledge to form a hypothesis. The ILP schema is as follows:

Positive examples + negative examples + background knowledge = hypothesis

(Muggleton, 1991)

In other words, ILP takes negative and positive input, combines this with the background knowledge and then creates a program that reflects all of the positive and none of the negative examples.

ILP uses an 'inverse resolution' strategy to draw conclusions from example input (Russell and Norvig, 2003).

4.4.2.1 Strengths of Inductive Logic Programming

Russell and Norvig (2003) identify three major advantages to using ILP over other inductive methods:

1. Rigorous approach to the general knowledge based inductive learning problem
2. Offers complete algorithms for inducing general first order theories from examples
3. ILP generated hypotheses are (relatively) easy for humans to understand

A rigorous approach to general knowledge is cited as an advantage over other similar techniques because of the ability of ILP to generalise relational models. For example consider the following 'family tree' problem, a popular predicate logic programming exercise. The following exercise is taken from Russell and Norvig (2003).

Consider expressing the concept "grandparent" as a predicate in decision tree learning. The predicates would need to be in pairs, such as:

Grandparent ({Mum, Charles})

When trying to represent example descriptions, the descriptions have to be very specific such as:

FirstElementIsMotherOfElizabeth({Mum, Charles}).

So the definition of 'Grandparent' becomes a series of specific cases, with no generalising.

In ILP generalising is possible by using background knowledge. If the background knowledge for the definition Grandparent contained the following:

Parent (x,y) <> [Mother (x,y) V Father (x,y)

Given this background knowledge, the definition of Grandparent could be reduced to:

Grandparent (x,y) <> [z Parent (x,z) ^ Parent (z,y)]

Thus the size of the hypothesis, in comparison to decision tree learning, is greatly reduced due to generalising using background information.

The second advantage is complete algorithms for inducing general first order theories from examples. Russell and Norvig state that this allows ILP to learn in situations where attribute based algorithms are hard to apply.

The final advantage of ILP is the manner in which hypotheses are output. Since hypotheses are output as understandable first order statements, the hypotheses can be scrutinised and understood relatively easily by humans.

Other methods, such as Neural Networks, do not offer such transparency. In fact, NN are often referred to as black box applications. Black box systems do not allow examination of the workings of conclusions which is a major disadvantage.

4.4.2.2 Limitations of Inductive Logic Programming

The greatest criticism of ILP comes from its inability to cope well with 'noise'. Noise is defined as *"when some of the data are incorrect"* (Russell and Norvig, 2003). In other words, noisy data contains incorrect examples.

Muggleton (1994) raises the issue of noise and ILP as a concern, further Muggleton identifies several other significant shortcomings in ILP which limits the applicability of the approach to real world situations. Muggleton (1994) states that the approach is *"too abstract"* in nature which makes it difficult to apply to the real world.

4.4.3 Genetic Algorithms (GA)

Genetic Algorithms (GAs) according to Russell and Norvig (2003) are defined as:

"A variant of stochastic beam search in which successor states are generated by combining two parent states, rather than modifying a single state"

Callan (2003) suggests that GAs are made up of two elements.

1. *Hypothesis encoding. Hypotheses need to be represented. Binary strings are typically used.*
2. *Objective function. An objective function is defined to evaluate the utility of a hypothesis. This evaluation returns a measure of how close the hypothesis is to a solution. This objective function is typically called fitness*

Further, evolutionary learning in GAs is achieved by the following six steps according to Callan (2003):

1. *Evaluate the fitness of each string using the objective function*
2. *Using a selection strategy, select the number of fittest strings*
3. *Apply genetic operators to generate new strings from those selected in step 2*

4. *Randomly mutate these new strings*
5. *Using a reinsertion strategy, generate the next population by replacing some of the existing strings with the new strings in steps 3 and 4.*
6. *If a solution is found, stop; otherwise return to step 1.*

Callan (2003)

However, as Mitchell (1998) notes, *" there is no rigorous definition of a genetic algorithm"*. Further, GAs have been likened to stochastic beam searches (Russell and Norvig, 2003) and state space searching (Callan, 2003).

4.4.3.1 Strengths of Genetic Algorithms

Luger (2005) states that an important strength of GAs is the parallel nature of their search. Luger (2005) draws comparisons between GA and NN in this vein, suggesting that a parallel computing capability is a great advantage in problem solving.

Mitchell (1998) and Cawsey (1998) state that an advantage of GAs as a machine learning method is the biological model it is based upon. Mitchell argues that the model of evolution is well suited to solving large complex computations.

Cawsey (1998) also argues that learning in GAs can be independent of the examples provided to it through the process of mutation.

4.4.3.2 Limitations of Genetic Algorithms

Criticisms of GAs to solve problems include the GAs providing solutions that are unrealistic or impossible. For example during the randomisation stage where individuals are given random combinations of 1's and 0's situations can arise where particular combinations in a particular context are invalid. (Mitchell, 1998)

This is also true during the crossover and mutation stages, some mutated individuals represent a combination of impossible states in certain contexts.

Another criticism of GAs is the high demand on resources, especially time. In particular the crossover stage, identifying which chromosomes to cross over in mutation, takes considerable time (Russell and Norvig 2003).

However, GAs have been used to solve complex problems such as the benchmark "Travelling salesman problem" (Borovska, 2006).

They have also been proven to be particularly complimentary to Artificial Neural Networks (Dokur *et al.,* 1997) especially when considering problems with little data available (Chann and Lippmann 1990, Mitchell 1998, Forman and Cohen 2004).

4.4.4 Neural Networks (NN)

Haykin (1999) defines Neural Networks (NN) as

A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two aspects:

1. *Knowledge is acquired by the network from its environment*
2. *Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge*

In other words NN take input from the 'environment' and adjust the networks synaptic weights to reflect the input/output characteristics of the data set.

The network learns by adjusting weights and observing the affect on the performance of the network until some pre-determined level of accuracy is attained. The way in which weights are adjusted varies depending on the learning rule used.

4.4.4.1 Benefits of Neural Networks inherited from the connectionist philosophy

Some of the benefits of NN come from the philosophy of the approach, NN are a "connectionist" philosophy.

The connectionist philosophy is based upon the assumption that "mental phenomena" can be represented by interconnected networks of simple processing units. The "mental phenomena" includes learning, therefore the philosophy believes that learning can be achieved via interconnected networks of simple processing units, i.e. NN.

The other major philosophy is the "symbolic" philosophy which assumes that learning is closely associated with language. Symbolic learning systems assume a finite alphabet of symbols that can manipulated to transform from one state to another. Learning is achieved by manipulating the alphabet into the desired form via a symbolic learning approach.

There are further advantages of a connectionist approach over a symbolic approach that are inherited by any connectionist method.

For example there is no need to assume a finite set of state combinations as in ILP and IES. Also there is no need for "background" knowledge as in ILP and IES. This significantly cuts down on the amount of programming needed, in fact NN are mostly self programming (Haykin, 1999).

Neural networks are particularly good at pattern matching and logical reasoning with applications in credit risk classification (Atiya, 2001, Yu *et al.* 2008), medical diagnosis (Zhou and Jiang 2003, Abbass, 2002) and marketing (Baesens *et al.*, 2002).

4.4.4.2 Strengths of Neural Networks

Haykin (1999) offers a more comprehensive list of benefits, he asserts there are nine major benefits of neural networks as follows:

1. *Nonlinearity*
2. *Input-output mapping*
3. *Adaptability*
4. *Evidential response*
5. *Contextual information*
6. *Fault tolerance*
7. *VLSI Implementations*
8. *Uniformity of analysis and design*
9. *Neurobiological analogy*

Haykin (1999)

Nonlinearity is the ability to create mathematical relationships between two or more variables that are not in a straight line (linear). In fact NN can either be linear or non-linear depending on the task at hand. This is seen as a great strength by Haykin (1999) and Principe *et al.* (2000) since it allows NNs to solve a wide variety of problems.

Input-output mapping is the process of 'supervised learning'. Supervised learning initiates a teacher-student relationship between the user and the NN. The user teaches the NN which combinations of inputs correspond to which classifications. This is cited as an advantage by Haykin (1999), Principe *et al.* (2000) and Stader (1992).

Adaptability in NN is the ability to adapt network weightings as conditions change in the environment, i.e. adjusting the network weights as different features are identified in the data (Haykin 1999, Principe *et al.* 2000)

Evidential response is the ability of a NN, especially when considering classification, to express the confidence of a classification or pattern (Haykin, 1999). This is useful for assessing the performance of the network and diagnosing when the network may not have learnt well.

Contextual information, i.e. knowledge representation is an advantage of NNs because of the interconnected structure of a NN (Haykin, 1999).

Fault tolerance is an ability of NN in the sense that under adverse conditions performance degrades gracefully rather than failing completely (Haykin, 1999). This is a viewpoint shared by Stader (1992) and Principle *et al.* (2000).

Further, this graceful degradation in performance also applies to 'noisy conditions'. This means that NN can perform with noisy or incomplete data sets which other methods such as ILP or IES have great difficulty with.

VLSI implementation is defined as Very Large Scale Integrated implementations, i.e. the *"ability to capture truly complex behaviour in a highly hierarchical fashion"*, (Haykin 1999). In other words NN have the ability to learn complex problems and present them in a highly structured hierarchical style.

Uniformity of analysis and design is a property that NN benefit from since all NN are built from the same simple information processing component, the artificial neuron (Haykin 1999). Haykin argues that it is therefore possible to share theories across different learning algorithms which can then be applied in other network structures.

Finally, neurobiological analogy is cited as a strength by Haykin (1999), Stader (1992) and Principle *et al.* (2000). Since NN are inspired by brains found in nature, the authors argue that this is *"living proof"* of the concept. A strong biological connection in Artificial Intelligence methods generally is seen as a great strength (Russel and Norvig 2003).

4.4.4.3 Limitations of Neural Networks

Whist the benefits of using NN are plentiful, Stader (1992) also identifies several important criticisms of the approach:

1. Problems of theoretical assessment
   (Provability)
2. Difficulties in designing neural network systems
   (Representation, Structure, Teaching)
3. Problems working with neural networks
   (Interpretation, Performance assessment, Scale)

Stader (1992)

Firstly the problem of provability i.e. being able to identify the specific rule that makes a trained network *correct*. Neural networks are often referred to as a 'black box' application, i.e. it may work correctly once trained but it is impossible to say exactly how it works. This is also true if training is halted mid-point, it is impossible to extract rules from the NN.

This raises some basic questions of neural nets, as Minsky and Papert (1988) noted: What can and can't it learn; How long should training take; How does network size affect performance.

Although both the Minsky and Papert's and Stader's papers were written in 1988 and 1992 respectively, these criticisms are still true today. Attempts have been made to extract rules from NN and thus 'open the black box' such as Setiono *et al.* (2000). However, this requires additional processing and is not considered a standard feature of most NN.

Difficulties in designing neural systems includes: Input and Network Structure

Issues concerning input are: Sufficient volume and quality of input data, also noted by Haykin (1999) and Principe *et al.* (2000). The volume of input is particularly stressed by Principle *et al.* (2000), although no benchmarks or heuristics are offered by the author. In contrast papers such as Plutowski *et al.* (1994) have demonstrated successful learning in neural networks with as little as 25 examples.

Problems arising from network structure such as setting the number of hidden layers in the network can also be problematic. There is no agreed heuristic for how many layers a NN should be for a given problem, advice is often vague on what hidden layers do, see Haykin (1999).

Further, the 'self programming' nature of NN, which is often cited as an advantage, removes control of network size from the operator.

Lastly Stader (1992) highlights the problems of working with NN, these include: Interpretation; Performance and Scale.

Stader (1992) defines interpretation of NN output to mean the process by which it arrives at a result. Stader argues that because NN are not symbolic it is difficult to rationalise the process of learning in NN.

In comparison traditional AI applications are symbolic, i.e. they manipulate symbols to solve rules. In symbolic AI it is simple to reproduce the sequence of events that manipulates symbols from the initial state to the end state. In this respect, this criticism is a similar argument to that of NN being 'black boxes'

Measuring the performance of a NN is identified as another difficulty. Stader states that since it is difficult to interpret what exactly is happening during learning, a means for assessing performance is particularly critical.

In classification problems, there are several means of assessing performance, Principe *et al.* (2000) suggests that the best measure of performance is a 'confusion matrix' which indicates the classification error of a network.

Logically, the best indicator of performance for a trained NN would be a blind test. The blind test consists of entirely unseen examples which are passed through the network, the networks response is then examined to give classification error on the unseen examples.

Finally the criticism of scale, i.e. the scale of time or some other resource that is consumed by NN operation is considered. Stader argues that because of hardware restrictions NN are inefficient, i.e. they take a *long* time to compute because of current hardware capabilities.

This is still true to some extent, however advances in computing power since 1992 when this paper was written have been substantial. For 'standard' NN time is not as critical as it was in 1992.

On the other hand, if NN are being used with Genetic Optimisation (GO) the amount of time needed to adequately learn can be several times that of a 'standard' network depending on the level of GO used.

The benefit of GO is a dramatic increase in accuracy (Mitchell, 1998) especially when training data is sparse (Chann and Lippmann 1990, Forman and Cohen 2004).

4.4.5 Conclusions on available machine learning algorithms

The conclusions drawn from the discussion on available machine learning approaches are as follows:

4.4.5.1 Inductive Expert systems

IES are dismissed as a viable machine learning approach since IES have now been largely superseded by ILP. Further, the techniques used in ILP are seen as advancement on those used in IES, making IES mostly redundant.

4.4.5.2 Inductive Logic Programming

Although ILP seems like an interesting and capable symbolic method, it is ruled out since it doesn't cope well with noisy data (Russell and Norvig 2003, Muggleton 1994).

Further, the usability of such a system is questioned by the abstract nature of implementing ILP (Muggleton, 1994). Expressions needed to code a model in ILP resemble those used in logic programming languages, such as Prolog, and will therefore be difficult for a non-IS professional to master.

However, the output which is easy for humans to understand would have been a valuable advantage (Russell and Norvig, 2003).

4.4.5.3 Genetic Algorithms

Although GA offer the strengths of parallel processing (Luger, 2005) and the strength of a strong biological analogy (Mitchell 1998, Cawsey 1998) standard GAs are ruled out.

This is due to inefficient time consumption (Russell and Norvig, 2003) and the potential to arrive at invalid solutions during randomisation and crossover (Mitchell, 1998).

However, when combining the power of Genetic Optimisation (GO) with another method such as NN, the benefits of GO can be realised without impacting severely on the amount of time an approach needs to learn.

In addition if GO is being used by another approach, say NN, the problem of invalid solutions is solved. The increase in time needed to learn when using GO is seen as a reasonable payoff for the increased accuracy it can offer.

One example of this is the GO of the input space of a NN, the GO greatly improves the accuracy of the NN without increasing the amount of time needed dramatically.

This has proved to be especially effective when only small amounts of training data are available for NN (Chann and Lippmann 1990, Mitchell 1998, Forman and Cohen 2004).

4.4.5.4 Neural Networks

Neural Networks are the most suited machine learning technique for EDM for several reasons.

Firstly the ability to deal with noise by gradual degradation is great strength of this approach (Haykin, 1999). Potentially this may allow EDM to be tolerant of user error such as BER.

Secondly NN are mostly self programming and self organising (Haykin, 1999), very little other than providing examples has to be done by the user. With EDM in mind, the self programming ability of neural networks may reduce the number of errors that arise from poor programming in spreadsheets.

The ability to give evidential responses is also cited as a major strength (Haykin, 1999). The ability to give evidential responses could potentially be used by the EDM modeller to determine the reliability of the EDM model.

Lastly the ability to generalise, although not exclusive to NN, is another great strength possessed by NN (Haykin, 1999)

Neural networks have been successfully applied to many similar problems that relate to decision support activities e.g. Bankruptcy prediction (Atiya 2001), Credit Risk (Yu *et al.*, 2008), Cardiac disease diagnosis (Azuaje *et al.* 1997), Diagnosis of breast cancer (Abbass, 2002), classification of level of return on stock investments Lueng *et*

al. (2000) and selection of trading strategies based on European stock prices (Andreou *et al.* (2008).

The common factor in the above applications of neural networks is that all of them are classification problems. Neural networks are particularly strong at classification problems judging by the number of applications in this area Zhang (2000) and Principe *et al.* (2000).

Since decision support neural net applications such as Atiya (2001), Yu *et al.* (2008) Azuaje (1997) Lueng *et al.* (2000), Abbass (2002) and Andreou *et al.* (2008) have been successfully applied to classification problems, it is reasonable to expect successful application to the decision support activities found in spreadsheets.

Although there are some serious criticisms of NN, some of these criticisms can be answered in part by using GO in conjunction with NN. Combining GO and Neural networks has shown to improve the learning ability of NN (Kim and Shin, 2007)

For example by genetically optimising the input space of the NN, one can relieve the issues of small training sets as identified by (Chann and Lippmann 1990, Forman and Cohen 2004).

Unfortunately, nothing can be done to alleviate the 'black-box' syndrome that NN exhibit. The 'black-box' syndrome is accepted as a necessary cost to using NN.

Costs on time and computing power (Stader, 1992), especially when using GO, can be mitigated to some extent through the use of relatively powerful hardware. In any case the issues identified by Stader (1992) regarding computer resource availability are somewhat outdated considering the large increase in computing power available on PCs since 1992.

The final conclusion is therefore to use NN with GO as the machine learning method to be used in EDM.

4.5 Neural Networks

This section explains the definition and workings of neural networks in more detail and how they can be applied to the novel approach, EDM.

4.5.1 Neural Networks overview

Haykin (1999) defines Neural Networks (NN) as follows

A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two aspects:

1. *Knowledge is acquired by the network from its environment*
2. *Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge*

A neural network consists of Artificial Neurons (also referred to as Processing Elements (PE)) arranged in a network. Artificial Neurons have the following four characteristics. See figure 4.3 for a diagram of an Artificial Neuron.

1. Input signals ($X_1 \ldots X_n$) which may come from the environment or other neurons
2. A set of weights attached to the input signals that describe connection strengths ($w_1 \ldots w_n$)
3. A threshold function that computes the neurons output state by determining how far above or below the threshold function a particular input pattern is. Typically the transfer function would be **sigmoid** as in the backpropagation algorithm however neural networks are not generally limited to only sigmoid.
4. Optionally, a learning rule that specifies how to adjust the weights for a given input/output pair.

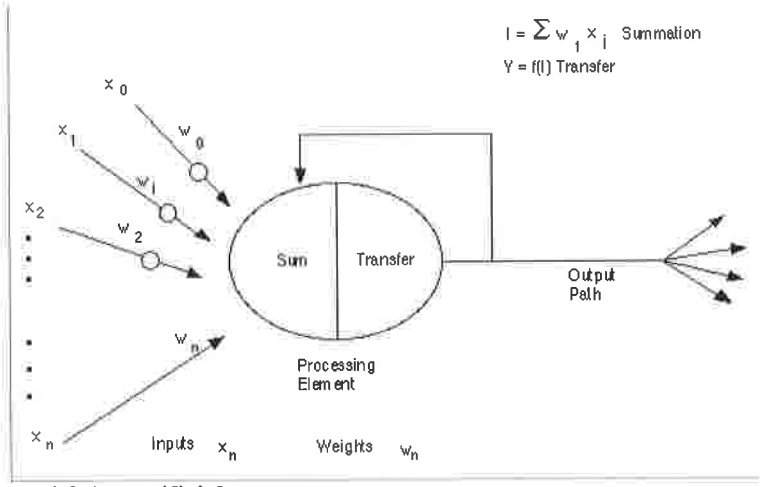

Figure 4.3 An artificial neuron

4.5.2 Learning in Neural Networks

Learning in NN can either be supervised or unsupervised (Haykin, 1999).

4.5.2.1 Supervised and unsupervised learning

Supervised learning uses a teacher-student relationship to adjust network weights to reflect the input. The teacher gives examples of attribute classifications to the NN, the NN then attempts to mimic the input output pattern of those attribute classifications (Haykin, 1999)

Unsupervised learning does not use a teacher, it is left to the network to distinguish the input/output features of the data. Unsupervised learning is more suited to exploratory tasks, such as data mining (Craven and Shavlik 1998 and Yang and Hamer 2007), for this reason unsupervised networks are not appropriate for EDM.

4.5.2.2 Supervised learning process

The mathematical process by which supervised NN learn is as follows:

1. A processing unit (an artificial neuron) takes a number of input signals, $X_1,...,X_n$ with corresponding weights $W_1...,W_n$, respectively.

2. These values are passed through the network to give an output which is then compared to the training set provided by the user.

3. The network then adjusts the weights in an attempt to mimic the input/output pattern of the training set.

4. This process is repeated until the network reaches some predetermined level of accuracy. This allows the network to become more and more accurate and hence the network learns the problem.

5. The neuron will only be fired if the threshold function (T) is satisfied and is governed by this equation: $X_1W_1 + X_2W_2 + X_nW_n > T$

4.5.3 Strategies for practical NN experimentation

In order to carry out experiments with neural networks and EDM one must decide how to implement such experimentation.

The basic choice faced is either building a bespoke application in a programming language, such as C++, or using a neural network development environment.

Both have advantages and disadvantages, using a bespoke application would allow perfect customisation. However, the development time would be significant, especially considering the likelihood that through the experimentation process, design details and features may have to be altered.

With a NN development environment, such as Neurosolutions (2007), the greatest advantage comes through the speed of development and the ability to alter design

elements as needed. The major disadvantage to this is that some control is surrendered to the package.

After much deliberation, it was decided that a development environment would facilitate experimentation more readily than a bespoke application. This flexibility is vital because of the adaptive nature of experimentation.

4.5.3.1 Choosing a NN development tool.

There are a variety of NN development tools available either as shareware or as a commercial product.

This software is either a simulation package or a component based package. Typically simulation packages are for developing and understanding the process by which NN learns and not necessarily for the application of NN to practical situations

Component based packages are aimed at applying NN to practical situations and offer 'plug-in' component functionality. The plug in components allows the user to change the make up and design of the network architecture to tailor a network to a particular need.

Clearly the component based network is more appropriate in the case of EDM than a simulation package since the aim of the thesis is to apply NN in real world situations and not to gain further understanding of the NN learning process.

After reviewing several component based packages, by means of downloading free trials and experimenting, Neurosolutions was chosen as the NN development environment.

4.5.3.2 Neurosolutions

According to Neurosolutions (2007) the neurosolutions package:

"...combines a modular, icon-based network design interface with an implementation of advanced learning procedures, such as conjugate gradients and back-propagation through time"

Neurosolutions offers two methods of developing NNs.

Firstly the 'breadboard' allows the user to pick and choose components to be included in the network architecture as needed, the user builds the entire NN from scratch.

The 'Neural Builder' allows the user to design NNs in the same manner as the breadboard but does not require the user to define synaptic connections between components and in that vein is more automated than the breadboard.

4.5.4 Conclusions on practical Neural Network experimentation

All experimentation was conducted using the neural network component package Neurosolutions.

The choice of a component based package over manually programming neural networks in an appropriate programming language is justified for the following reasons.

Firstly a component based package was chosen over a simulation package because the aim of the thesis is to test a novel idea in practical real situations. Neural network Simulation packages are typically designed to help improve the understanding of the learning process, component based packages are designed to be applied to practical situations.

Secondly using a component based package allows greater flexibility than if one were programming a neural network. In a component based package changing a variable in the neural network design is a matter of clicking the mouse, a network programmed in C++ or Java would be harder to maintain.

After evaluating several neural network component packages Neurosolutions was chosen to implement the neural network experiments.

Neurosolutions offers a wide range of neural network architectures and learning algorithms, therefore consideration must be given to the NN design for experimentation.

4.6 Neural Network design

Since there are many configurations of neural network available, choosing the best set up is a complex task.

Firstly the network architecture must be considered, the network architecture is the design and interconnection of neurons in the network. In addition, the network architecture may, but not always, dictate the learning algorithm.

Further the 'hidden layer' depth must be set, the 'hidden layer' describes a number of hidden neurons in a network architecture.

The use of GO must also be considered, since GO can be applied to several different parameters in the NN.

However, before the details of the neural network design are discussed and evaluated, the following section provides some definitions and some background theory on how the 'universe' of examples is related to training, cross validation, testing and blind data sets.

4.6.1 The Universe, training, cross validation, testing and blind testing sets

The purpose of this sub-section is to define some commonly used terms and to explain the relationships that exist between them.

Figure 4.4 shows the relationship between the training set, testing set, cross validation set, blind testing set and the 'universe' of examples.

The 'universe' of examples contains every possible combination of attribute classifications for a given problem.

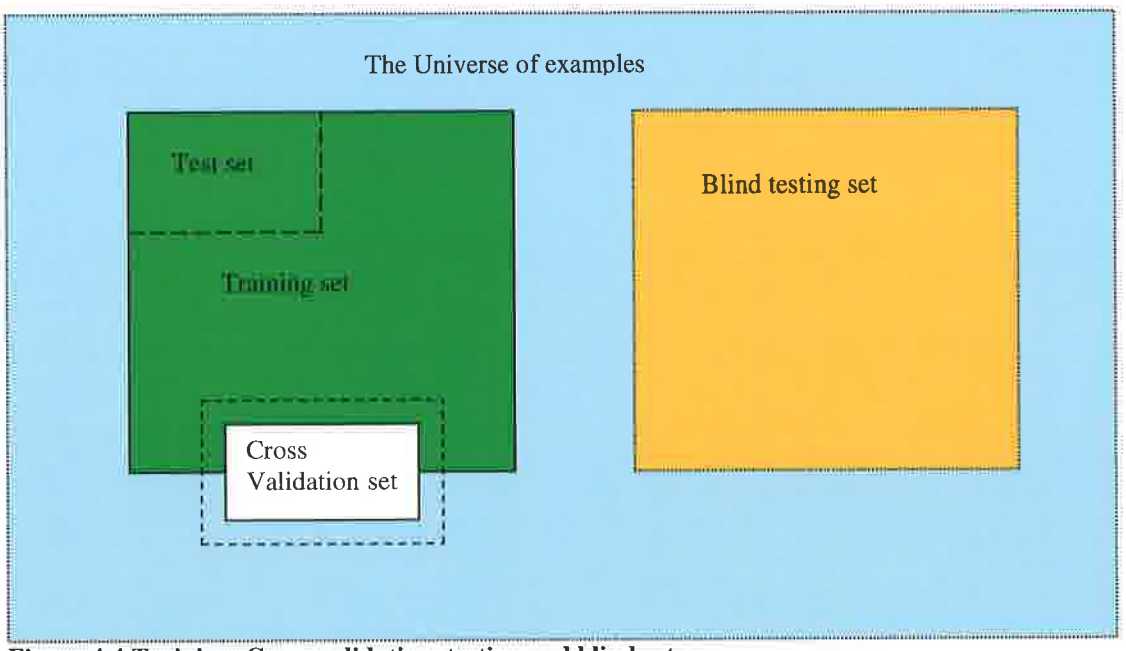

Figure 4.4 Training, Cross validation, testing and blind sets

The training set (T set), which is a subset of the universe, is used to train the neural network. Included in the training set are the test and cross validation sets. These provide two important means to measure performance in the trained network.

The training set, a subset of the universe, comprises of a finite number of attribute classifications. The training set is used to train the neural network and is also used to measure the internal performance of the network, i.e. performance ignoring the wider

universe of examples. This is calculated using Mean Squared Error (MSE), MSE is a statistic that calculates error in the network.

The Cross Validation (CV) set is a subset of the training set and of the universe. The CV set is comprised of examples from the training set and examples taken from the universe. These unseen examples are used to asses the networks ability to generalise to the unseen universe also using MSE.

The test set is a subset of the training set and is used in the testing phase of neural network design. Typically the test set has far fewer examples than the training set and is used to give a quick impression of the networks ability. However, since the test set is a subset of the training set, the examples contained in it are 'known'. In other words the network has already used these examples to train the network, thus any performance indication must be used carefully since the test set ignores unseen examples.

The blind testing set contains unseen 'blind' examples used as an absolute measure of accuracy. The network classifies the blind set, having already been trained using the training set, and the results are manually checked for error. Blind testing is the best means of determining the absolute accuracy of the data.

Once the blind testing set results have been checked, the classification error or the classification accuracy can be calculated. Classification error is the percentage of incorrect classifications, classification accuracy is the percentage of correct classifications.

4.6.2 Network architecture

The network architecture, sometimes referred to as 'network paradigm', relates to the structure of neurons in the NN and the relationship between neurons and the learning algorithm.

The network structure can be 'simple' i.e. a 'feed-forward single layer neural network'. This NN consists of only input and output neurons, see figure 4.5.

Feed-forward networks pass information forward, from left to right, starting at the input layer and finishing at the output layer. Most importantly feed-forward networks pass information only in one direction.

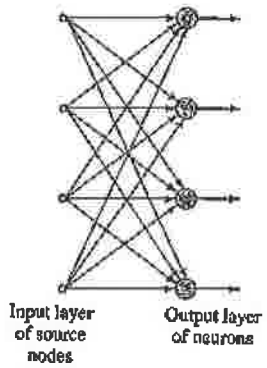

Figure 4.5 Feed-forward Single layer network

The network architecture can also include more layers to give a 'feed-forward multi-layer' network that incorporates hidden layers, see figure 4.6.

Multi-layer networks can either be partially connected or fully connected. In fully connected networks all neurons are interconnected by synaptic connections between layers, where some of these synaptic connections are not present we have a partially connected network.

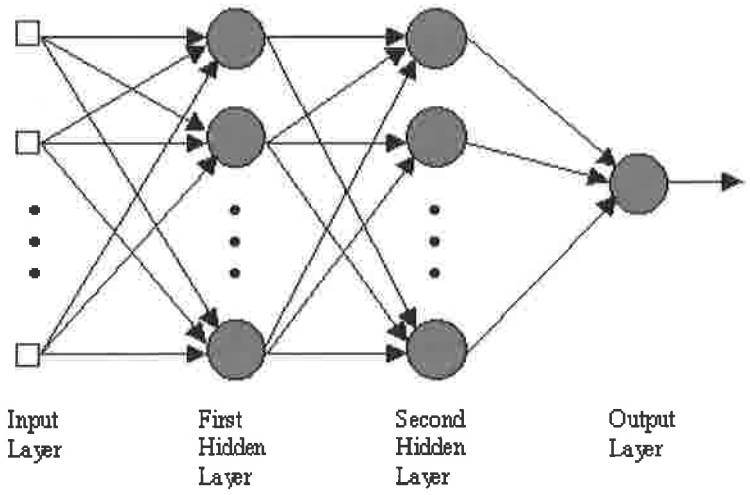

Figure 4.6 Feedforward multi layer network

The alternative to a feed-forward network is a recurrent NN, which uses at least one feedback loop. This feedback loop passes values either from the output layer neurons back to input layer neurons or neurons can have a self feedback loop. Recurrent networks can be either single or multi layer.

4.6.3 Learning algorithm

There are many different algorithms for learning in NN. Since EDM essentially takes input and classifies it, the learning algorithm must be able to classify. Any algorithms using unsupervised learning as part of the learning process were discarded (As discussed in section 4.5.2.1).

This effectively leaves a choice between Multi Layer Perceptrons (MLPs) and Support Vector Machines (SVMs).

4.6.3.1 Support Vector Machines (SVMs)

The SVM is a specialised learning algorithm and network architecture for classification.

Haykin (1999) states that

"The support vector machine has the ability to solve pattern-classification problems close to the optimum for the problem at hand"

In other words the SVM is a very good classifier and is often regarded as a superior classifier in comparison to MLPs.

"The absence of local minima from the above algorithms [Support Vector Machines] marks a major departure from traditional systems such as neural networks,..."
Shawe-Taylor and Cristianini (2002)

And further:

"Classical learning systems like neural networks suffer from their theoretical weakness, e.g. back-propagation usually converges only to locally optimal solutions. Here SVMs can provide a significant improvement."

Rychetsky (2001)

However, there are some major criticisms of SVMs.

"However, from a practical point of view perhaps the most serious problem with SVMs is the high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks."
Horváth (2003)

In other words using a SVM puts high demands on computing resources, since the amount of resources available is limited to standard PC, using SVMs may present major difficulties.

This point is echoed by Burgess (1998) who states:

"Perhaps the biggest limitation of the support vector approach lies in choice of the kernel...a second limitation is speed and size, both in training and testing."
Burgess (1998)

Other than criticising the method by which SVM distinguish feature patterns, Burgess notes that training and testing SVMs takes considerable time.

4.6.3.2 Multi Layer Perceptrons (MLPs)

The learning algorithm used in MLPs is the back-propagation learning algorithm, MLPs are a popular classification choice (Haykin, 1999)

Haykin (1999) notes that MLPs have been used to solve some difficult classification problems and are a common choice amongst researchers since it is based upon the popular error back-propagation algorithm

Further Haykin (1999) identifies three major advantages to the back-propagation learning algorithm:

1. *"Artificial Neural Networks that perform local computations are often held up as metaphors for biological neural networks"*

Here Haykin (1999) argues that a strong biological connection between artificial NN and biological NN is a strong advantage

2. *"The use of local computations permits a graceful degradation in performance due to hardware errors, and therefore provides the basis for a fault tolerant network design"*

Here Haykin (1999) argues that because of the ability of MLPs to gracefully degrade performance is more constant when error is introduced. Here Haykin (1999) uses the term *"Hardware errors"*, this means failure in the hardware of the network architecture, not failure of the computer hardware.

Further, the benefit of graceful degradation extends beyond hardware error to noisy data sets. In fact noise in the training data has been shown to improve the learning ability of MLPs (Jun *et al.,* 2002)

Haykin's (1999) final point is

3. *"Local computations favour the use of parallel architectures as an efficient method for the implementation of artificial neural networks"*

Here Haykin notes that MLPs and the back-propagation algorithm are computationally efficient, i.e. they are relatively efficient at consuming computing resources when compared with other techniques.

The major criticism of MLPs and the back propagation algorithm is convergence to local minima (Haykin, 1999). Convergence to local minima is defined as the tendency for the learning algorithm to get 'stuck' in a local minima rather than finding the global minima. Back-propagation is particularly at risk from this since it uses a "gradient-descent" approach, i.e. evaluate the neighbouring data points to search for global minima. Figure 4.7 shows how a learning algorithm can get 'stuck' in local minima.

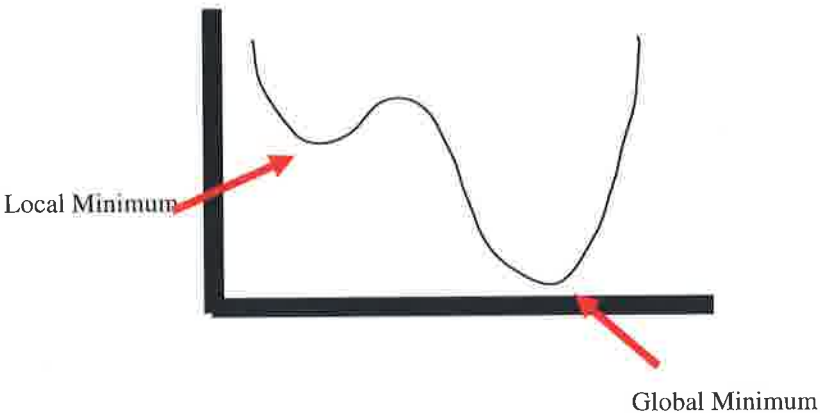

Figure 4.7 Local and Global minima

The local minima in figure 4.7 has higher neighbouring data points around it, this may cause the back-propagation algorithm to consider the local minima to be the global minima. Where this arises, the optimum search solution is not found.

However, 'momentum' can be used to mitigate the risk of local minima (Principe *et al.,* 2000). Momentum acts as inertia for the back-propagation algorithm, prompting the search of further data points beyond what appears to be the global minima. By doing this, local minima can be escaped and the true global minima can be found.

In Neurosolutions 'momentum' is a standard feature of the MLP and back-propagation algorithm.

4.6.3.3 The chosen learning algorithm

Choosing between these two algorithms presents a difficult decision. Whilst SVM can achieve a close to optimal solution (Rychetsky, 2001), the amount of time to learn is reported to be excessive (Burgess, 1998). According to Haykin (1999) MLPs and the back-propagation algorithm can achieve good solutions relatively quickly but do suffer from local minima.

Considering that the problem of local minima can be mitigated (Principe *et al.* 2000) the question over which to use comes down to how long a SVM takes to learn a problem.

Since this is not adequately described in the literature a small comparison experiment was designed. A simple Male or Female crab classification problem was used as a benchmark. The crab classification exercise and training data is taken from Principe *et al.* (2000)

A standard SVM and a standard back-propagation MLP were trained using the male or female data set. The only recorded parameters were the amount of time it took to learn the problem and the classification accuracy of the network.

The training set consisted of 100 examples, the amount of time taken by the MLP was 4 minutes and 26 seconds for which it achieved a 89.7% accuracy. The amount of time taken by the SVM was 13 minutes and 11 seconds for which it achieved 93.4% accuracy.

Given the amount of time taken versus the accuracy, the SVM takes considerably longer to learn than the MLP. Further, considering there was substantial experimentation with EDM, the more efficient option was more appropriate.

Therefore the chosen learning algorithm for use in the EDM experiments was the MLP and back-propagation algorithm.

4.6.4 Hidden layer depth

'Hidden layers' are a feature of MLP networks, hidden layers are layers between the input and output layer, see figure 4.8

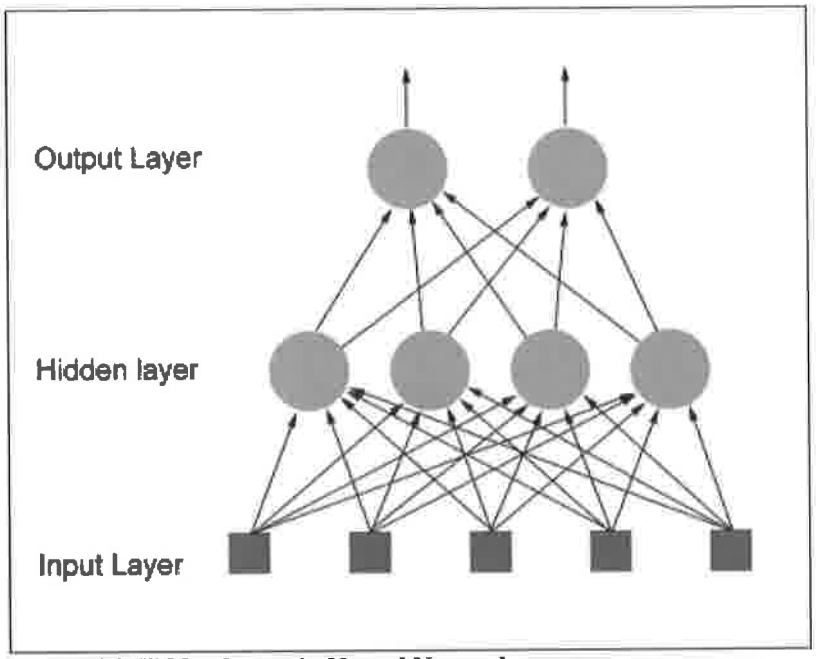

Figure 4.8 Hidden layers in Neural Networks

The purpose of hidden layers, as Haykin (1999) notes, is to *"intervene between the internal input and network output in some useful manner"*.

Further Haykin (1999) goes onto assert *"By adding one or more hidden layers, the network is enabled to extract higher order statistics from the data"*.

So therefore the number of hidden layers directly affects the networks ability to learn from the data.

In the chosen paradigm, MLP, there has to be at least one hidden layer but information on how many a network should have is unclear. The following set of quotes offers some guidance and some discrepancy:

"A rule of thumb is for the size of this [hidden] layer to be somewhere between the input layer size ... and the output layer size ..."
(Blum, 1992).

"you will never require more than twice the number of hidden units as you have inputs" in an MLP with one hidden layer"
(Swingler, 1996).

"How large should the hidden layer be? One rule of thumb is that it should never be more than twice as large as the input layer."
(Berry and Linoff, 1997).

"Typically, we specify as many hidden nodes as dimensions [principal components] needed to capture 70-90% of the variance of the input data set."
(Boger and Guterman, 1997)

From the above evidence it is clear that hidden layers are beneficial but how many seems to be contentious.

Since there is no research consensus, the number of hidden layers is automatically controlled by the software, dynamically pruning or adding hidden layers as necessary.

4.6.5 Genetic optimisation

Genetic optimisation in neural networks is often applied to the input data to optimise input pairs for the network. Use of genetic optimisation is generally seen to be positive for accuracy (Dokur e*t al.,* 1997)

It has also been shown to dramatically improve learning accuracy in networks using small amounts of training data (Chann and Lippmann 1990, Forman and Cohen 2004)

Considering that the use of genetic optimisation is beneficial to learning and that the use of genetic algorithms can relieve the problem of small training sets, genetic optimisation was used in all experiments.

4.6.6 Performance indicators

Measuring the performance of the network, i.e. the accuracy of the network can be achieved through several different means. There are statistical measures such as Mean Squared Error (MSE) that reveal the closeness of fit between the desired output and the actual network output, MSE is a modified version of chi squared.

There are also methods that take into account the shape of the learning curve and how that shape indicates the performance of the network.

Confusion matrixes can also be employed to measure performance in classification type problems. The confusion matrix shows the number of correct and incorrect classifications derived by comparing the network output with the 'known' output.

Lastly blind testing is considered as a means to "acid test" the networks reliability of output.

4.6.6.1 Mean squared error

When considering the fit of an estimator to an actual amount, the error relates to the difference in these two numbers. The Mean Square Error (MSE) is thus a measure of how well an estimated function fits the real data. For each and every observation it looks at the difference between the estimation and the amount being estimated. These differences are squared and then averaged (i.e. summed together and divided by the total number of observations), resulting in the MSE figure for the estimator in question. This concept is defined in Equation 4.1 below:

$$MSE(y) = \frac{\sum_{i=0}^{N}\sum_{j=0}^{P}\left(d_{ij} - y_{ij}\right)^2}{NP}$$

Equation 4.1 MSE

Where

P = Number of output processing elements

N = Number of exemplars in the data set

y_{ij} = Network output for exemplar i at processing element j, $i = 0, ..., N$ and $j = 0, ..., P$

d_{ij} = Desired output for exemplar i at processing element j, $i = 0, ..., N$ and $j = 0, ..., P$

The misfit or difference between the estimated and the actual can either be due to randomness or an indication that the estimator does not fit the data optimally.

4.6.6.2 Learning curves

Learning curves can be a useful tool in determining if the NN has learnt well. The learning curve is simply a plot of the MSE values for the T and CV sets against the number of iterations the NN has completed in the training phase.

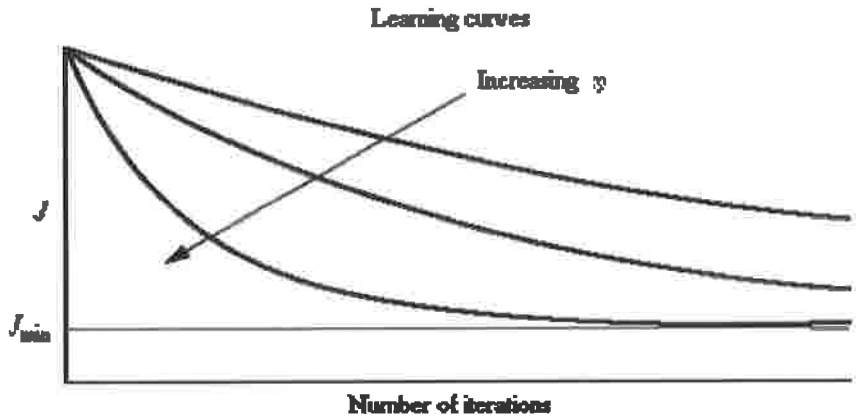

Figure 4.9 Examples of learning curves

Figure 4.9 presents some examples of learning curves, J and J_{min} relate to the MSE and MSE_{min} (the optimum MSE value). The different curves could either be different training sets such as cross validation and training or they could show different attempts at learning the problem.

Critical to determining the performance of a network is the shape of the learning curve. The examples in Figure 4.9 indicate the network has learnt well, the curves are

smooth and predictable. An erratic learning curve, i.e. a curve that is not smooth, suggests the network has not learnt well.

However, whilst the shape of the learning curve indicates if the network has learnt well or not, it does not show to what extent the network has learnt. Therefore this measure must be used with others for interpretable results.

Learning curves, on their own, cannot guard against local minima and should be used in conjunction with other performance measures.

4.6.6.3 Confusion Matrixes

A confusion matrix is defined as:

"...a simple methodology for displaying the classification results of a network. The confusion matrix is defined by labelling the desired classification on the rows and the predicted classifications on the columns"
(Principe *et al.*, 2000)

The confusion matrix simply tabulates the number of correct and incorrect classifications as either a real number or a percentage in a matrix.

Table 4.2 shows a confusion matrix expressed using percentages, as can be seen in this example the network gave an imperfect result, the network misclassified male as female 11% of the time and female as male 5% of the time

	Male	Female
Male	89	11
Female	5	95

Table 4.2 Simple confusion Matrix example

This means of evaluating performance is arguably more useful than any of the other measures since this method explicitly expresses the accuracy of the network.

The size of the matrix is defined by the number of classifications in the training set. If the data had three classifications, the corresponding confusion matrix would be 3 by 3, rather than 2 by 2 as table 4.2 shows.

4.6.6.4 Generalisation to unseen data performance

Whilst there is no particular test for measuring the networks ability to generalise, one can use confusion matrix output, MSE and blind testing.

By comparing the difference in CV and T values from the confusion matrix it is possible to detect degrading generalisation. An ideal result would show the CV and T values close together which indicates that the performance of the training set and the cross validation set are similar.

This in turn suggests that the network is able to generalise to 'unseen' examples (the CV set) with approximately the same accuracy as the 'seen' examples (the T set).

The same effect can be observed in the learning curve. If the CV and T MSE values are close together see figure 4.10, it is a good indication that the network can generalise with a similar accuracy achieved in the training set.

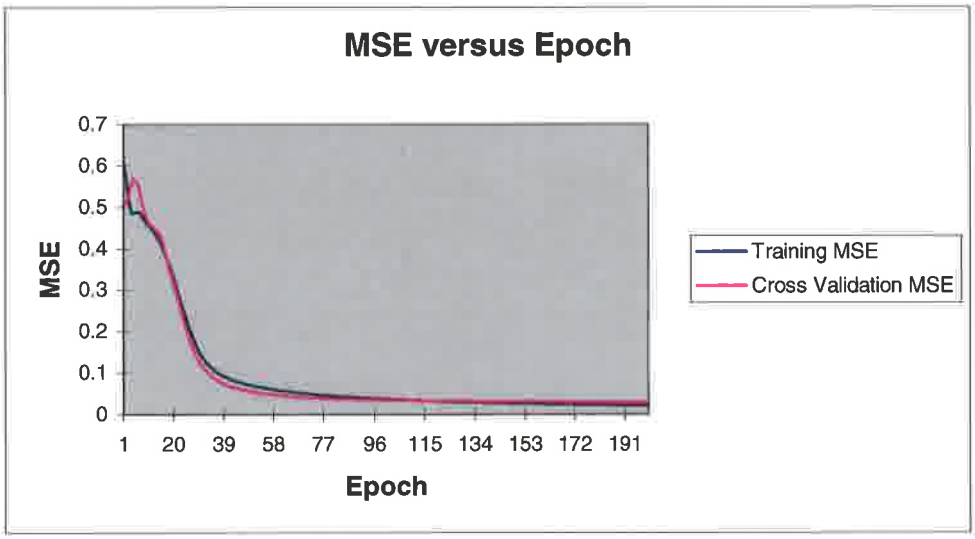

Figure 4.10 Example T and CV MSE values

The further apart the T and CV values are, the worse the ability of the network to generalise.

4.6.6.5 Blind testing sets

Blind testing sets refer to the blind testing of a network with examples that were not used in training process whose output is known. By running blind sets through the network and comparing the results with the known values, one can measure how well the network will perform on entirely new examples.

This is a similar idea to a confusion matrix but in this method only unseen examples are used, a confusion matrix uses examples the network has trained on. Anecdotal evidence from the neural network community suggests that this type of testing is the best measure of true performance.

Once this process of manually checking the blind test data is complete, the classification accuracy and the classification error can be calculated.

The 'blind' examples are passed through the network and the blind test results are output by the network. The output comes in the form of un-normalised probability. Once the un-normalised probability is output, these probabilities must be manually checked to see if the network is assigning the highest probability to the correct classification.

All testing undertaken by the network, blind or otherwise, is output as 'un-normalised probability'. Un-normalised probability falls between the values -0.05 and 1.05 as opposed to 'normalised' probability which falls between 0 and 100. The reason for this, uncovered in private communication with the developers of the software, is that:

> *'...using the values 1 and 0 often cause problems during simulation. Changing these values to 'un normalised' probability remedies this issue'*

One might consider altering these values to 'normal' probability but the range of un-normalised probability is greater than 'normal' probability. This causes some difficulty in converting the values meaningfully. Further the minimum number is a minus value which adds to the difficulty of conversion.

Since the actual values are not important, it is the highest probability for a given example that counts, using un-normalised probability doesn't raise any significant difficulties in interpreting the results.

4.6.7 Conclusions on Neural network design

For clarity, the following conclusions were reached when discussing potential neural network designs to be used in experimentation.

4.6.7.1 Network architecture and learning algorithm

Multi Layer Perceptrons (MLPs) were chosen over Support Vector Machines (SVMs) as the chosen neural network architecture since relative accuracy, tolerance of user error and resource consumption are considered superior in MLPs.

The learning algorithm, in this instance, is defined by the network architecture. The learning algorithm is the *'error back-propagation algorithm'* based on gradient descent.

4.6.7.2 Hidden layers

Owing to a lack of consensus in the literature, three hidden layers were used initially but the neural network software was able to prune or add hidden layers dynamically as necessary (as section 4.6.4).

4.6.7.3 Genetic optimisation

Owing to the accuracy benefits when data sets are small, genetic optimisation of the input data set was used in all experiments (as section 4.6.5).

4.6.7.4 Performance indicators

Whilst all the performance indicators were considered, more emphasis was be placed on the use of confusion matrixes and blind testing as a means of presenting results. This was simply because they were both easier to interpret and give a clearer indication of performance over both MSE and learning curves.

MSE and learning curves are more useful diagnostics tools than a means of evaluating performance. Therefore these methods were primarily used when the network was being trained.

4.7 Summary of neural network design

For clarity, the configuration of the neural networks to be used in the experiments is as follows:

1. The Neural networks were developed in the neural network development environment Neurosolutions.
2. The neural networks used in the experiments were Multi Layer Perceptrons (MLP) using the supervised learning technique of the *error back-propagation algorithm*
3. The number of hidden layers was set by the neural network dynamically
4. The input for the MLPs was genetically optimised using a genetic algorithm to optimise the input space
5. Learning curves and MSE were used to assess the learning process of the MLPs but were not used as an absolute measure of accuracy

6. Absolute accuracy was determined by blind testing sets and the use of confusion matrixes.

4.8 Advantages and disadvantages of using neural networks

This section aims to summarise the main advantages gained from using neural networks as the means of implementing example-giving.

Main advantages to using neural networks for implementing example-giving:

- Mostly self-programming
- Noise tolerant (to some level)
- Less sensitive to data when generating models (when compared to other inductive techniques)
- Evidence based confidence levels in results
- Relatively efficient computation

The above advantages gained from using neural networks to implement example-giving additionally resolve some if the disadvantages of example-giving (section 3.9) such as BER and bias issues in the creation of examples.

Disadvantages to using neural networks for implementing example-giving

- Insufficient volume of examples implies a variable result.

Further questions regarding example-giving implemented with neural networks are dealt with in the next chapter.

4.9 Conclusions of chapter

In this chapter we considered the external validity or generalisation of results based on the tightly defined problem domain see section 3.1.1.

Considerations on how example-giving could be practically implemented to deal with complexity are contained in section 4.3.6. After evaluating the various methods, machine learning was considered a better option for modelling large complex problems.

Conclusions on the various machine learning algorithms available are contained in section 4.4.5. After evaluating the strengths and weaknesses of the various machine learning techniques available, Neural Networks were chosen as the means to implement example-giving. This is due to several attributes: Graceful degradation (including noise); Self organisation and self programming and a particular strength in generalising.

An additional benefit of Neural Networks is their generally accepted ability to classify robustly (Rumelhart and McClelland 1988, Mitchell, 1999 and Principe *et al.* 2000). Considering the problem domain in section 3.1.1 this ability will be of great benefit in coping with complexity.

Section 4.6.7 contains specific details of neural network design. Firstly a neural network component package was chosen over a simulation package. Secondly the network architecture and learning algorithm chosen were MLPs and the *error back propagation algorithm* since they offer better relative accuracy, training time, error tolerance and can incorporate genetic optimisation.

Section 4.7 provides a reference summary of the design features discussed in section 4.6.

This chapter has discussed and evaluated several different methods that could have been used to implement example-giving for decision support spreadsheet models.

After considering what could be used to implement example-giving, machine learning was eventually chosen as the best suited to the domain (decision support spreadsheets). Further, neural networks were the preferred choice and a detailed design for those neural networks has been developed.

This chapter in conjunction with chapter 3 fully satisfies objective 2:

"Based upon the literature review, consider an alternative modelling technique for the reduction of error in decision support spreadsheets"

Therefore the last objective to be considered is objective 3:

"Investigate, develop, test and evaluate the proposed novel approach"

This final objective is the subject of the next two chapters.

5.0 Experiments in machine learning

5.1 Chapter overview

Section 5.1 introduces the EDM design aspects for experimentation conducted in this chapter. Sections 5.2 and 5.3 discuss the design and results respectively of an experiment examining the effect of varied size training sets. Sections 5.4 and 5.5 discuss the design and results of the increased complexity experiment which examines the effect on NN performance as the complexity of the training sets is increased. Sections 5.6 and 5.7 discuss the variance and sensitivity experiments which investigate how sensitive training sets are and also the variance present in EDM learning. Sections 5.8 and 5.9 discuss the EDM with noise experiments designed to measure the effect of noise on the EDM learning process and discover what the effect of noise is on performance of EDM. Section 5.10 provides a summary of the experimentation contained in this chapter. Section 5.11 summarises the relative advantages gained from implementing EDM with NN. Section 5.12 concludes the chapter.

5.1.1 Introduction to experimentation

The four experiments contained in this chapter deal with three critical areas: Performance and training set size; Performance and problem complexity, the reproducibility of performance and the effect of noise on performance.

First consider the justification for these experiments. These four areas of experimentation have been chosen because they are relevant to assessing the performance of EDM when using neural networks as the means of implementation. Moreover the four experiments are critical success factors when considering the definition and specification of the problem domain see section 3.1.1. For example, if EDM needed an excessive number of examples or was unable to cope with medium to high complexity or produced an unreliable result or was significantly affected by noise in the data set, then the usefulness and applicability of EDM to decision support spreadsheets would have been significantly reduced.

Experimentation on performance and training set size relates to the disadvantage identified in section 4.8 (insufficient examples implies a variable result). Experimentation is necessary since the true number of examples needed for a satisfactory performance is unknown.

Experimentation on performance and problem complexity is in a similar vein to that of performance and training set size. The effect of problem complexity has an unknown effect on performance, thus experimentation is necessary.

Section 4.8 cites that one advantage of neural networks is they are less sensitive to data when generating models (when compared to other inductive techniques). However, it is unclear how reproducible any given result is, i.e. the variance in results is not known. Therefore experimentation on the reproducibility of results is needed to fully understand this.

Further, section 4.8 cites another advantage of neural networks is the ability to deal with noise without adversely affecting performance. Again although this is stated in the literature, it is unclear how noise affects performance of EDM, therefore experimentation was needed to understand this.

The aims for this chapter were:

1. Measure the effect on EDM performance when reducing the training set size.
2. Measure the effect on EDM performance when increasing the complexity.

3. Measure the variance in EDM performance to assess the reproducibility of results.
4. Measure the effect on EDM performance when noise is introduced to the training set.
5. Use the results from the four experiments to determine if EDM (example-giving combined with Neural Networks) is a practically viable means for modelling decision support spreadsheets as detailed in section 3.1.1

5.1.2 The design of neural networks to be used in all experimentation

In the following experiments a standard design is followed for all neural networks used in experimentation, see section 4.7.

Further, performance was measured in accordance with section 4.6.7.4.

A standard design was used in all experiments since it allowed fairer comparison than if each network were individually tailored.

5.1.3 Generation of training sets used in experiments

Other than where stated, training sets were generated by random number generation in Microsoft Excel. The only exception to this is in the last experiment which uses both a randomly generated set and a tailored set.

Details on how the various data sets are constructed are contained in each respective section on each experiment.

5.1.4 The definition of EDM in this chapter

EDM is defined in this chapter as example-giving implemented using neural networks to produce 'models' based on the example input given by a user.

This definition is provided to save writing 'example giving implemented with neural networks' *ad nauseam.*

Although it is important to note that the focus of this chapter is on the performance of EDM using neural networks, neural networks are only one option for implementing example-giving.

5.2. Reduced training set experiment

Principe *et al.* (2000) state that the training set is of critical importance to the effective learning of the neural network.

> *"The size of the training set is of fundamental importance to the practical usefulness of the network. If the training patterns do not convey all the characteristics of the problem class, the mapping discovered during training only applies to the training set."*

Here Principe suggests that if the training set only covers a portion, not all, of the problem, the resulting solution will only represent the portion trained on.

Further Principe *et al.* (2000) comments on the size of the training set, i.e. how many examples there are in the training set.

> *"Another aspect of proper training is related to the relation between training set size and number of weights in the NN. If the number of training examples is smaller than the number of weights, one can expect that the network may "hard code" the solution, i.e. it may allocate one weight to each training*

example. This will obviously produce poor generalization (i.e the ability to link unseen examples to the classes defined from the training examples). We recommend that the number of training examples be at least double the number of network weights."

Therefore a potential difficulty arises with neural networks that could affect EDM. Basically put, one must obtain *sufficient number* of examples from the user to train the network adequately and allow effective generalisation.

According to Principe *et al.* (2000) the consequence of fewer examples in a training set is poor generalisation to unseen examples.

However, some published work suggests that neural networks can be trained with as little as 25 examples. Plutowski *et al.* (1994) demonstrated that with 25 examples plus or minus 2, they could adequately train a neural network to predict the Mackey-Glass time series. The Mackey Glass time series is a chaotic time series and is seen as a benchmark in the time series prediction amongst the neural network community.

In summary there seems to be some discrepancy on how many examples are needed in the training set for the network to learn properly. Therefore the first experiment takes a simple problem and measures the effect on performance as the number of examples in the training set is steadily reduced.

This experiment determines the effect on accuracy when the training set of a sample problem is steadily reduced down and beyond the 25 examples limit suggested by Plutoski *et al.* (1994).

5.2.1 Experimental aim

The aims of this experiment were:

1. Discover the minimum size of training set needed to adequately implement EDM

2. Measure the effect of reducing training set size on performance at intervals of 750, 500, 250, 100, 50, 25, 20, and 15 examples in a training set.
3. Assess the impact of the findings on EDM.

5.2.2 The sample problem

The sample problem is task 4 in chapter 3. The requirements of this task were to classify a student's grade based upon two inputs, coursework and exam. The average mark was taken from exam and coursework and was used to compute the grade.

Although this was a simple task, complexity was tested in another experiment. The aim of this experiment is to measure the effect of reduced sets on accuracy.

5.2.3 Generating the training sets

Training sets were generated for the nine different groups (750, 500, 250, 100, 50, 25, 20, 15 and 10) using a random number generator in Microsoft Excel. Each set is born from the same parent population, i.e. the training sets are different only in the size of the set and not the examples contained in them. This is a fairer test condition since it rules out differences in training sets effecting accuracy.

A certain amount of manual processing was then required to ensure that the randomly generated examples were valid. Table 5.1 contains a small excerpt from the 750 example training set.

Example number	CW (coursework)	EX (exam)	AVER (average)	Fail	Pass	Merit	Distinction
1	91	56	74	0	0	0	1
2	58	9	34	1	0	0	0
3	39	92	66	0	0	1	0
4	32	15	24	1	0	0	0
5	64	73	69	0	0	1	0
6	13	85	49	0	1	0	0
7	58	69	64	0	0	1	0
8	79	44	62	0	0	1	0
9	83	16	50	0	1	0	0
10	73	76	75	0	0	0	1

Table 5.1 Training set excerpt

5.2.4 Dividing the examples into input and desired classes

Once the examples have been generated and loaded into the neural network, they must be divided into input and desired classes. Further any redundant information is removed from the training set since leaving redundant information in slows the learning process unnecessarily.

Table 5.2 shows how the data is divided into input (green), output (red) and redundant (amber) data. The redundant data, example number and average, are deleted or ignored by the neural network.

Example number	CW (coursework)	EX (exam)	AVER (average)	Fail	Pass	Merit	Distinction
1	91	56	74	0	0	0	1
2	58	9	34	1	0	0	0
3	39	92	66	0	0	1	0
4	32	15	24	1	0	0	0
5	64	73	69	0	0	1	0
6	13	85	49	0	1	0	0
7	58	69	64	0	0	1	0
8	79	44	62	0	0	1	0
9	83	16	50	0	1	0	0
10	73	76	75	0	0	0	1

Table 5.2 Training set divided into input, desired and redundant

Once the data has been assigned as either input or desired the neural network is ready to learn from the data.

For the experimentation, this process of generating examples and dividing them into input and desired classes is repeated for each of the nine different sized training sets.

5.3 Results of reduced set experiment

The results of this experiment are expressed as the blind testing results only and not the confusion matrix results. This is simply for the sake of presentation, full confusion matrix result and summary confusion matrix results can be found in Appendix B.

Figure 5.1 shows the classification accuracy of the trained network when presented with the blind testing set

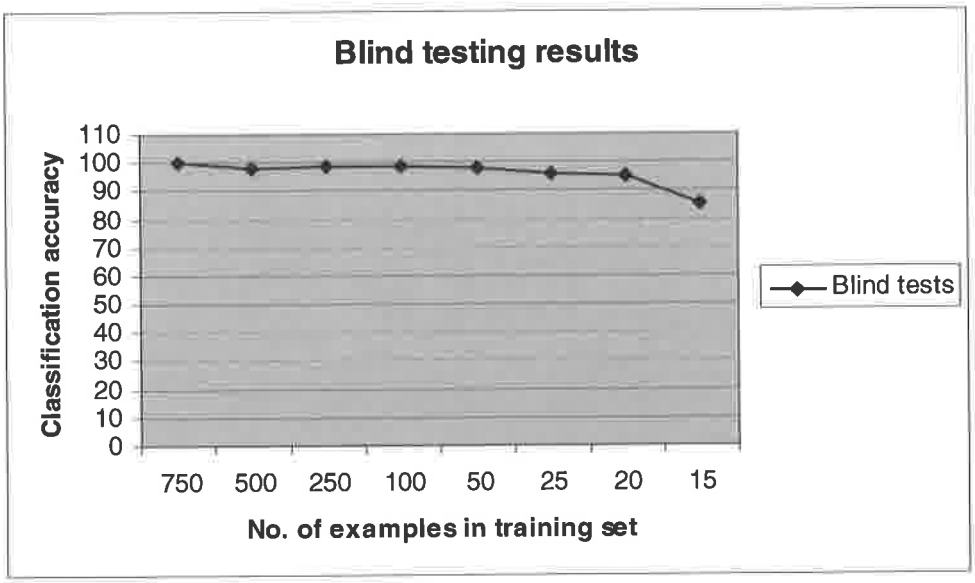

Figure 5.1 Blind testing classification accuracy

As can be seen in figure 5.1, the classification accuracy does not drop below 90% until the 15 example training set is tested. In fact the accuracy does not drop below 95% until the 20 and 15 example sets are tested.

In the case of the 15 example training set, the accuracy drops to 85% from 94% for the previous example set (20 examples). This is a significant fall in accuracy and

considering the unusual behaviour observed from the 15 example set (see appendix B, section 1) this fall in accuracy is expected.

However, the networks accuracy holds up surprisingly well considering the low numbers of examples provided in the training set and evidence presented in Principe *et al.* (2000).

However, Plutoski *et al.* (1994) demonstrated adequate learning with as few as 25 examples using genetic optimisation as Chann and Lippmann (1990) and Forman and Cohen (2004) advocate

5.3.1 Conclusions of reduced set experimentation

The aims of this experiment were to

1. Discover the minimum number of examples needed to train a classification network.
2. Reveal the effect on accuracy as training set size is reduced.

5.3.1.1 Aim 1

As the literature suggests, Plutoski *et al.* (1994), the minimum number of examples needed to train a classification neural network *adequately* is 25.

This conclusion is formed on the basis that the blind testing results show that for 25 examples the network accuracy remains at 95% or above, see figure 5.1. If one were seeking a higher accuracy, such as 99%, at least 100 examples would be required.

5.3.1.2 Aim 2

The effect of reducing the training set size is a decrease in the classification accuracy, see figure 5.1.

The confusion matrixes contained in section 1 of appendix B show a critical drop in performance when the training set size is reduced to 15. When the 15 example set was blind tested, see figure 5.1, it showed a considerable drop in accuracy from 94% to 85%

5.4 Increased complexity experiment

The second experiment is concerned with measuring the effect on accuracy when increasingly complex problems are modelled using EDM.

5.4.1 Experiment aim

The aim of this experiment is

1. Discover the effect of increased classification complexity on EDM performance.
2. Investigate if there is a practical limit to classification complexity that limits the use of EDM.

5.4.2 The sample problems

Since the aim of this experiment was to measure how well EDM performs in problems of increasing complexity, the sample problem must was made sufficiently complex.

According to Prechelt (1994) increased complexity of problem is achieved by increasing the classification complexity. The classification complexity refers to the number of classifications in a problem. Increasing the number of classifications in the problem increases complexity in the training set and in turn increases the process of learning the classifications.

Therefore the problems contained in this experiment became increasingly more complex by increasing the classification complexity of each of the training sets.

The sample problem started with the minimum number of classifications, two and extended to training sets with 10 classifications.

The sample problems for this experiment were based upon the previous experiment, the grade classification system. However, since the number of classifications increases as the experiment progresses, some invention of grades was necessary.

5.4.3 Generating the training sets

As in the first experiment all sets (training, testing, cross validation and blind testing) were randomly generated using Microsoft Excel.

However, in this experiment since the number of classes change for each sub experiment, the training sets cannot be born from the same parent population, i.e. each training set for each experiment must be unique since it is impossible to re-use training sets because of increased classification complexity.

Thus these experiments lose the benefit of using the same parent population, i.e. each data set used for each experiment is different. This means that potentially differences in the composition of training sets could affect the performance of the neural network and skew the results of the experiment. To mitigate this each training set contained 100 examples which minimised the effect of differences in training set composition.

Further, the choice of 100 examples for the training allowed observation of the effect classification complexity had on performance, rather than the effect of reduced training sets on performance.

Training sets with classes of 2, 3, 4, 5, 6, 7, 8, 9 and 10 were created. This required some invention of grades so that the number of classifications for each set could be achieved. See table 5.3 for a description of the training sets used.

Sub experiment no.	No. of classifications in task	Logical rule	Classification details
1	2	Average of CW and EX	Fail <= 39, >= 40 Pass
2	3	Average of CW and EX	Fail <= 39, Pass >= 40 & < 55, Merit >= 55
3	4	Average of CW and EX	Fail <= 39, Pass >= 40 & <55, Merit >= 55 & <70, Distinction >= 70
4	5	Average of CW and EX	<= 39 Fail, Pass >= 40 & < 55, Merit >= 55 & < 70, Distinction >= 70 & <80, Commendation >= 80
5	6	Average of CW and EX	<= 39 Fail, Pass >= 40 & < 55, Merit >= 55 & < 70, Distinction >= 70 & <80, Commendation >= 80 & < 90 Excellence >= 90
6	7	Average of CW and EX	< 35 Fail, Compensate Pass >= 35 & <40, Pass >= 40 & < 55, Merit >= 55 & < 70, Distinction >= 70 & <80, Commendation >= 80 & < 90 Excellence >= 90
7	8	Average of CW and EX	<=20 Re-sit module < 35 & > 20 Fail, Compensate Pass >= 35 & <40, Pass >= 40 & < 55, Merit >= 55 & < 70, Distinction >= 70 & <80, Commendation >= 80 & < 90 Excellence >= 90
8	9	Average of CW and EX	<= 10 Redo year <=20 & >10 Re-sit module < 35 & > 20 Fail, Compensate Pass >= 35 & <40, Pass >= 40 & < 55, Merit >= 55 & < 70, Distinction >= 70 & <80, Commendation >= 80 & < 90 Excellence >= 90
9	10	Average of CW and EX	<= 10 Redo year <=20 & >10 Re-sit module < 35 & > 20 Fail, Compensate Pass >= 35 & <40, Pass >= 40 & < 55, Merit >= 55 & < 70, Distinction >= 70 & <80, Commendation >= 80 & < 90 Excellence >= 90 & <95 University award >= 95

Table 5.3 Details of experiment data sets

5.4.4 Dividing the examples into input and desired classes

The examples are divided up into input and desired columns as described in section 5.2.4. As stated above, since the classification complexity increases, i.e. the number of classifications in the training set increases for each experiment, it is not possible to use a single parent to create all of the training sets.

5.5 Results of increased complexity experiments

In this experiment, for the sake of presentation, only the classification accuracy data will be included in the main text. The other ten confusion matrix graphs are included in section 2, appendix B.

5.5.1 Blind testing results

Figure 5.2 shows the classification accuracy of the trained network when presented with blind sets with increasing classification complexity.

As can be seen from the graph, there is no significant change in classification accuracy until the classification complexity reaches 8 classes.

With 8 classes the classification accuracy drops below 95% and with 9 classes the classification accuracy drops below 90%.

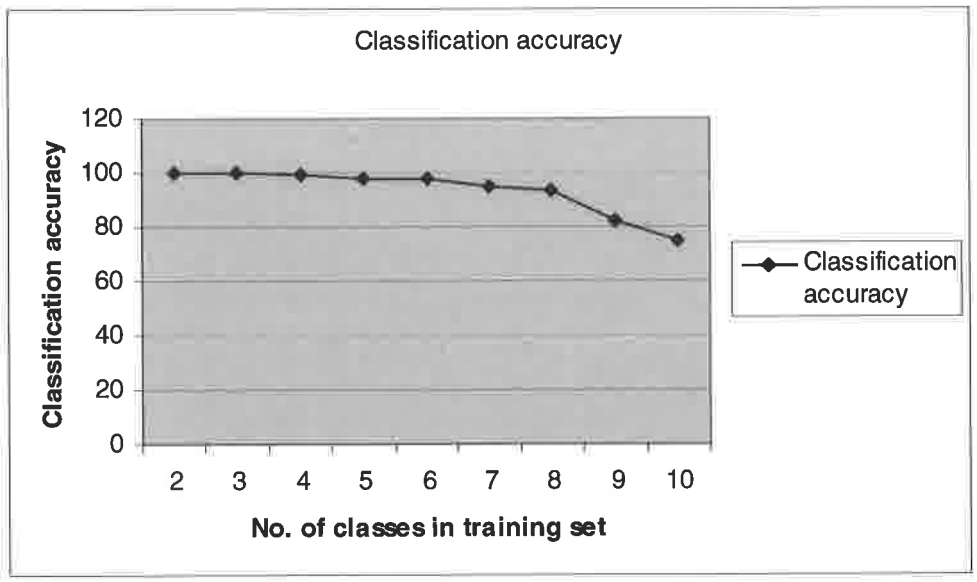

Figure 5.2 Blind testing classification accuracy

Therefore in this case, the network performs satisfactorily with training sets with a classification complexity of 7 or less provided there is a training set of 100 examples.

The Neural Network literature shows that the more complex the scenario the more data is needed as shown by (Prechelt, 1994) in solving the travelling salesman problem and the Soybean disease classification problem, both complexity benchmarks in the Neural Network community.

Further *"the backpropagation algorithm is computationally efficient"* Haykin (1999), in other words the algorithm can reasonably cope with increasing complexity.

5.5.2 Conclusions of increased complexity experiment

The aims of this experiment were to:

1. Discover the effect of increased classification complexity on accuracy using an EDM approach
2. Investigate if there is a practical limit to classification complexity that would prevent the use of an EDM approach.

5.5.2.1 Aim 1

The effect of increased classification complexity is the reduced performance of EDM once the classification complexity exceeds 7 classifications.

5.5.2.2 Aim 2

Results indicate that after more than 7 classifications causes the classification accuracy to drop below the 95% level.

This would seem to suggest that the practical limit of complexity is 7 classifications. However, neural networks have been shown to complete classification problems with far greater complexity than that of this increased complexity experiment.

An example of this is the Soybean disease classification problem which is considered a benchmark in the neural network classification community. The Soybean benchmark problem has 34 inputs, 19 classes and 683 examples in the training set (Prechelt, 1994).

This problem has been solved many times over the past few decades using a variety of techniques (Brent 1991, Sexton and Dorsey, 2000, Wang *et al.* 2004) including the use of MLPs similar to those used in this experiment (Sexton and Dorsey, 2000).

This research suggests that classification complexity could be irrelevant in small problems. It is the researcher's hypothesis that if there are enough good examples evenly distributed between all the classes in the training set, the classification complexity becomes sidelined and the volume of examples is the critical factor.

So this would suggest that problems with a larger classification complexity merely require a larger training set to adequately cover all classes and learn the problem.

This is not reflected in the results of this experiment possibly because of the manner in which the training sets are generated. Since the training sets are generated randomly, the number of examples per class is therefore random. Some classifications may have 20 examples and some may have 2. This theory was tested in the next experiment.

5.6 Variance and training set sensitivity

During the course of experimentation with neural networks it was observed that even with identical conditions, the learning process varies slightly and has a *small* effect on the overall performance of the network

Therefore the next experiment in this chapter aims to measure the variance present when using neural networks to implement example-giving. These results helped determine if variance had a significant impact on the usefulness on EDM.

Further, this experiment tested if allocating an equal number of examples to classes in the training set improved the accuracy of the network as discussed in section 5.5.2

5.6.1 Experiment aims

The aims of this experiment were:

1. Discover the level of variance present by repeating multiple simulations using the same training data and neural network design
2. Assess the impact of the variance on the usefulness of EDM
3. Test the theory that if there are enough 'good' examples spread evenly across classifications in the training set, the performance of the network increases over the performance in a randomly generated training set.

5.6.2 The sample problem

The sample problem in this experiment drew on the findings of the reduced training set experiment and the increased complexity experiment.

There were two purposes behind the training sets used in this experiment. Firstly the training sets were used to measure the variance present in multiple repeat simulations and secondly to determine if tailored training sets offered a significant performance advantage.

In this experiment two training sets were used: the control and treatment groups.

The treatment group had a 'tailored' training set and were run 10 times to measure the variance in the results. Tailored training sets are those that are not created randomly, i.e. the training set is designed to have an equal number of examples per classification.

The performance of the treatment group, the tailored training set, was compared to that of the control group, the random training set, to determine if tailoring the training set offered a significant performance advantage over the randomly generated control set.

The control group used the original randomly generated training set from the increased complexity experiment. This was also run 10 times to measure any variance.

5.6.3 The treatment group training set

The treatment training set divided the 100 examples available evenly into 7 classes. Since 7 does not divide into 100 equally, 98 examples (divisible by 7) were used in the training set for the treatment group.

5.6.4 The control group training set

The control group training set used 98 randomly generated examples with at least 1 example per class. The choice of 98 examples was due to the fact that the treatment group can only have 98 examples and in the interests of fairness, this condition was imposed on the control group too.

5.7 Results of variance and sensitivity experiment

The variance and sensitivity were measured by comparing two measures of performance.

The classification accuracy, derived from blind testing, and MSE provided by the neural network were both used to measure variance in multiple simulations.

Variance was measured by comparing the changes in classification accuracy and MSE values over 10 simulation runs, i.e. the same experiment was run 10 times to record any differences in classification accuracy and MSE between them.

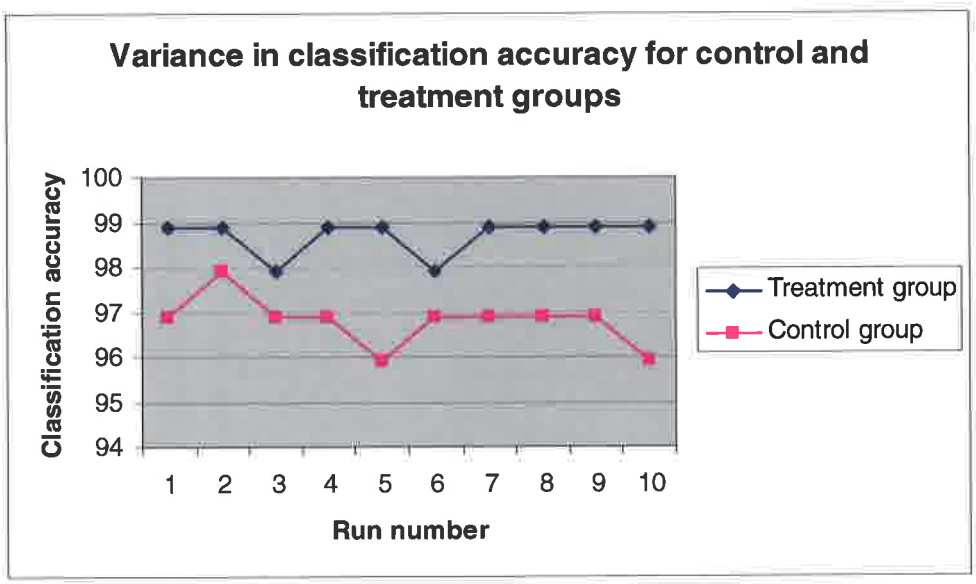

Figure 5.3 Variance in classification accuracy for treatment and control groups

Figure 5.3 shows the variance in classification accuracy between the treatment and control groups.

The treatment group showed slightly higher classification accuracy than the control group, the average increase was 1.9%.

The increased classification accuracy for the treatment group is not significant enough to prove the hypothesis that tailoring training sets are superior to randomly generated ones. However, it does consistently show some improvement in accuracy which suggests tailoring is beneficial to accuracy but not significant.

The treatment group showed a minimal variance, in 80% of the data the classification accuracy was 97% in the other 20%, the classification accuracy dropped to 96%. The range was 1%, this variance was not significant enough to raise concerns about the reproducibility of the result.

The control group variance was higher than the treatment group, in 70% of the data the classification accuracy was 95%, in 10% of the data the classification accuracy was 96% and in 20% of the data the classification accuracy was 94%. The range was 2%, this was not significant enough to raise concerns of reproducibility.

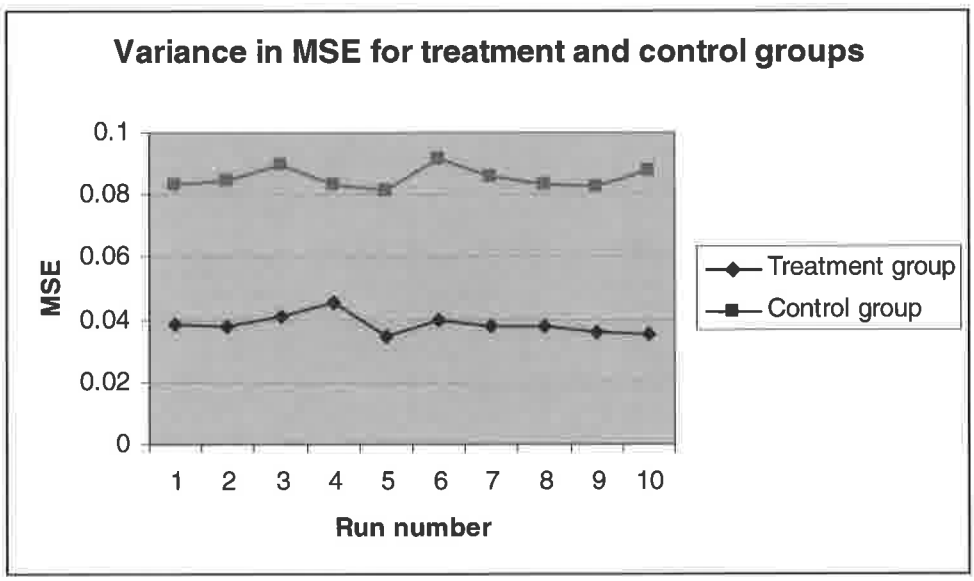

Figure 5.4 Variance in MSE for treatment and control groups

As in figure 5.3, the MSE values in figure 5.4 showed better performance for the treatment group than control group. However, this difference was not significant enough to conclude that the tailored training set offered significant advantage.

The range of MSE values for the 10 simulation runs in the treatment group was 0.01049. Such a small range did not raise concerns of reproducibility.

The range of MSE values for the 10 simulations runs in the control group was 0.010168. Again this range was small enough not to raise concerns of reproducibility.

5.7.1 Conclusions of variance experiment

The aims of this experiment were:

1. Discover the level of variance present by repeating multiple simulations using the same training data and neural network design
2. Assess the impact of the variance on the usefulness of EDM
3. Test the theory that if there are enough 'good' examples spread evenly across classifications in the training set, the performance of the network increases over the performance in a randomly generated training set.

5.7.1.1 Aim 1

The experiment showed that there was variance in MSE and classification accuracy for the treatment and control group across all 10 simulation runs.

The variance found in the classification accuracy and MSE for the treatment group were 1% and 0.01049 respectively.

The variance found in the classification accuracy and MSE for the control group were 3% and 0.010168 respectively.

5.7.1.2 Aim 2

The level of variance in classification accuracy and MSE detected in 10 simulation runs for both the control and treatment groups all using the same respective conditions was judged to be insignificant. Such low values do not raise concerns of reproducibility.

Therefore this variance was not significant enough to impact on the usefulness neural networks as a means to implement example-giving.

5.7.1.3 Aim 3

The data in figures 5.3 and 5.4 goes some way to supporting the notion that tailored training sets provide superior accuracy in comparison to those training sets that are randomly generated. On average the tailored training set increased accuracy by 1.9%

However, the difference in classification accuracy and MSE were judged to be too small to decisively conclude that tailored training sets were superior to randomly generated training sets.

They do appear to be beneficial for accuracy but it is not significant enough to be decisive.

5.8 The performance of EDM with noise experiment

The performance of EDM under noisy conditions must be considered because of the possibility that users will make simple mistakes and effect of unavoidable human factors such as BER (Panko, 1999).

Regardless of the care taken, BER will be present in any task humans undertake. BER levels depend on the task at hand but have been reported to be around 5% for complex programming type tasks (Panko, 2007).

With this in mind, it is important to examine how noise levels (user error levels) might affect the usefulness of a neural network based implementation of example-giving.

5.8.1 Experiment aims:

The aims of the experiment were:

1. Measure the effect on EDM performance when noise is introduced into the training set
2. Determine if the effect discovered in aim1 significantly impacts on the viability of neural networks as a means of implementing example-giving.

5.8.2 The experiment task

In this experiment the selected task used 100 examples covering 4 classifications. This task was the same one used in the first experiment in this chapter, the reduced set experiment.

In order to gain a benchmark level of performance, the first simulation run used a training set with 0% noise.

The subsequent simulation runs used training sets that have 5%, 10%, 15%, 20% and 25% noise in them. Adding 'noise' to the data set was achieved by introducing erroneous training examples into the set.

All simulation runs used training sets born from the same parent, i.e. the training sets were identical bar the level of noise contained in each respective set.

5.9 Noise experiment results

Figure 5.5 shows the effect of noise on the classification accuracy. Confusion matrix results are included in appendix B for the sake of presentation.

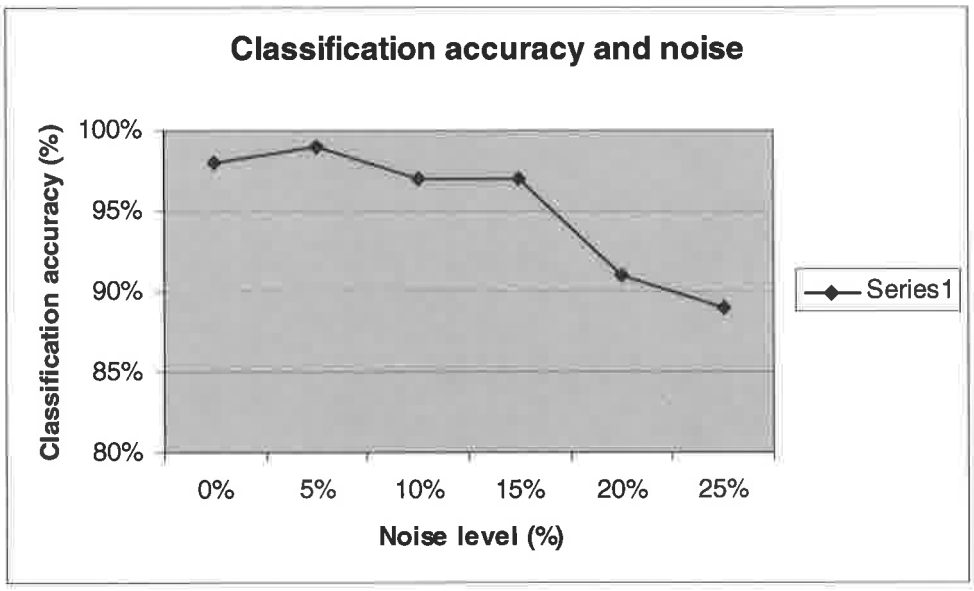

Figure 5.5 The effect of noise on classification accuracy

As can be seen in figure 5.5, generally speaking the more noise introduced to the training set the lower the classification accuracy. However, the rate at which classification accuracy drops was unexpectedly low and with 5% noise the classification accuracy improved.

Considering neural network literature this is not an unexpected result as Principe *et al.* (2000) state:

> *"Trained with noisy inputs it will eventually learn the important parts of the input pattern then after training if the input patterns that are noisy or incomplete [it] will reconstruct the correct image because it has enough information from the input pattern to correctly construct the output pattern" p232*

This notion that noise assists learning is also discussed in Rumelhart and McClelland (1988) and is reported recently by several authors (Hasegawa and Umeno 2008, Zhang 2007). The process of adding noise to training sets to improve validity response is known as "additive noise"

Using training sets with up to 15% noise (i.e. 15% of the training set contained erroneous examples) caused little to no effect on classification accuracy.

This is a significant finding because the trained NN is able to maintain 97% classification accuracy even though 15% of the training set is incorrect.

5.9.1 Conclusions on noise experiments

The aims of this experiment were:

1. Measure the effect on EDM performance when noise is introduced into the training set
2. Determine if the effect observed in aim1 significantly impacts on the viability of neural networks as a means of implementing example-giving.

5.9.1.1 Aim 1

The effect on performance when noise was introduced was a reduction in classification accuracy. This reduction in classification accuracy is only present once at the 20% noise level, i.e. 20% of the training set is erroneous.

However, a small amount of noise (no more than 5%) appeared to improve the classification accuracy, see figure 5.5

5.9.1.2 Aim 2

The findings of aim 1 do not severely impact on the usefulness of neural networks as a means of implementing example-giving. On the contrary the results reveal a strength, being able to tolerate 15% error is advantageous.

5.10 Conclusions on performance experimentation

The aims for this chapter were:

1. Measure the effect on EDM performance when reducing the training set size.
2. Measure the effect on EDM performance when increasing the complexity.
3. Measure the variance in EDM performance to assess the reproducibility of results.
4. Measure the effect on EDM performance when noise is introduced to the training set.
5. Use the results from the four experiments to determine if EDM (example-giving combined with Neural Networks) is a practically viable means for modelling decision support spreadsheets as detailed in section 3.1.1

These aims were achieved by conducting four performance experiments to gain an understanding of the performance strengths and weaknesses of neural networks as a means of implementing example-giving.

What follows is a brief summary of results and conclusions followed on by a more detailed break down of each aim and conclusions drawn from those aims.

5.10.1 Summary of findings

Using neural networks, EDM can learn adequately from 25 examples simple problems.

More complex problems (up to 7 classifications) can be learnt with 100 examples in the training set. The literature suggests that neural networks can classify more complex problems at the expense of gathering more data.

The reproducibility of results has been shown to be reliable, exhibiting a small variance in performance over 10 identical simulations.

Further up to 15% of these examples can be incorrect before the classification accuracy of EDM starts to fall, i.e. EDM can tolerate up to 15% error. In addition a 5% noise level improves classification accuracy.

5.10.2 The effect of reduced training set size on performance

The effect on performance with smaller training sets was achieved via three sub aims:

1. Discover the minimum size of training set needed to adequately implement EDM
2. Measure the effect of reducing training set size on performance at intervals of 750, 500, 250, 100, 50, 25, 20, and 15 examples in a training set.
3. Assess the impact of the findings on EDM.

5.10.2.1 The minimum size of training set needed to adequately implement EDM

The minimum number of examples in a training set is 25 achieving a classification accuracy of 95% or above which considering statistical significance is the acceptable minimum. With fewer examples the classification accuracy drops below 90%.

By increasing the number of examples, the classification accuracy can be increased to 100%.

5.10.2.2 The effect of reducing training set size on performance

The effect of reducing the training set size is a reduction in performance, i.e. the fewer examples the poorer the classification accuracy and confusion matrix results. See figure 5.1.

5.10.2.3 Assess the impact of the findings on EDM.

The impact of these results on EDM are as follows, firstly the smallest size of training set that yields satisfactory performance is 25. Using 25 examples one can achieve 95% classification accuracy.

By using training set sizes of 50 and 100 examples per set, the classification accuracy can be increased to 98% and 99% respectively.

The uncovering of the minimum number of examples needed in the training set allows one to understand EDMs limitations and also how EDM is likely to perform under certain conditions.

5.10.3 The effect of increased complexity on performance

The effect on performance with increasing complexity was achieved via two sub aims:

1. Discover the effect of increased classification complexity on EDM performance.
2. Investigate if there is a practical limit to classification complexity that limits the use of EDM.

5.10.3.1 The effect of increased complexity on performance for EDM

Assuming that 100 good examples are contained in the training set, EDM can perform sufficiently with up to and including 7 variables. Beyond 7 variables, performance drops below the minimum required level.

As figure 5.2 shows, once the classification complexity is beyond 7 classes the classification accuracy drops below 95%. As discussed earlier, 95% is the acceptable minimum

5.10.3.2 Evaluate the practical limit of complexity for EDM

Given the performance data in figure 5.2, one could suggest the practical limit is 7 classes.

However, further experimentation showed that with a tailored training set it was possible to elevate the classification accuracy on average by 1.9%, section 5.6 expands on this in more detail.

In addition, the literature suggests that if there is enough data to learn from, classification problems of great complexity can be learnt satisfactorily.

One such example is the Soybean disease classification problem which is considered a benchmark in the neural network classification community. The Soybean benchmark problem has 34 inputs, 19 classes and 683 examples in the training set (Prechelt 1994).

The main difference between the Soybean problem and the increased complexity experiment is the size of the training set. The Soybean problem uses a training set of 619 examples, the increased classification complexity training sets only contain 100 examples.

This suggests that increased complexity requires an increase in the training set size to learn satisfactorily. The same may be true for EDM, greater classification complexity requires larger training sets.

In conclusion, the practical limit of complexity for EDM is 7 classes for 100 example training sets.

5.10.4 Variance in performance (the reproducibility of results)

The aim of assessing the variance in performance was achieved by three sub aims:

1. Discover the level of variance present by repeating multiple simulations using the same training data and neural network design
2. Assess the impact of the variance on the usefulness of EDM
3. Test the theory that if there are enough 'good' examples spread evenly across classifications in the training set, the performance of the network increases over the performance in a randomly generated training set.

5.10.4.1 The level of performance variance present in multiple identical simulations

The level of performance variance present in multiple simulations using identical conditions was measured using classification accuracy and MSE.

Variance in classification accuracy and MSE, see figures 5.3 and 5.4, was minimal and not deemed to be statistically significant enough to raise concerns of reproducibility.

5.10.4.2 Assess the impact of variance on EDM

Since the variance is not statistically significant, there is no adverse impact on EDM, although classification accuracy may vary by a maximum of 3%.

5.10.4.3 Tailored versus randomly generated training sets

There is some evidence to suggest that tailored training sets can increase classification accuracy in complex problems but the difference observed in this experiment is too small to prove or disprove such a theory.

Figures 5.3 and 5.4 show the treatment group had a slight accuracy advantage over the control group in both classification accuracy and MSE, on average this increase is 1.9%

It is possible that with further experimentation this theory could be more thoroughly tested. However, since evidence in the literature suggests that complex classification problems can be solved with an adequately sized training set, further experimentation would not assist the thesis.

5.10.5 The effect of noise on performance

The aims of this experiment were:

1. Measure the effect on EDM performance when noise is introduced into the training set
2. Determine if the effect discovered in aim1 significantly impacts on the viability of neural networks as a means of implementing example-giving.

5.10.5.1 The effect of noise on performance

The effect of introducing noise into the training sets is a reduction in classification accuracy once the level of noise reaches 20% or higher, although 5% noise improves classification accuracy. Figure 5.5 contains results from the noise experiment.

5.10.5.2 The impact of performance under noise on the viability of EDM

The impact of the noise experiment on the viability of EDM is marginal. As can been seen in figure 5.5, no noticeable effect was present until the noise level reached 20% or more. On the contrary, maintaining a level of 97% classification accuracy where 15% of the training set is erroneous is a great strength.

As previously mentioned a noise level of 5% appears to be beneficial, i.e. 5% noise actually improves the classification accuracy.

5.10.6 The impact of the experimental findings on the usefulness of EDM for decision support spreadsheets

Considering experimental aim 5, section 5.1.1:

Use the results from the four experiments to determine if EDM (example-giving combined with Neural Networks) is a practically viable means for modelling decision support spreadsheets as detailed in section 3.1.1

The evidence gathered from the experimentation (Sections 5.3, 5.5. 5.7 and 5.9) suggests that EDM is a practically viable method for implementing decision support spreadsheets. This is due to several factors: EDM does not require an excessive number of examples to learn (Section 5.3); EDM can cope with complexity which can be enhanced by allowing additional time or providing additional examples (Section 5.5); The EDM results show very little variation – it is reliable (Section 5.7) and EDM performs well with noisy data sets (Section 5.9).

5.11 Advantages and disadvantages of EDM implemented with neural networks

This section will briefly summarise the advantages and disadvantages gained from using neural networks to implement example-giving.

Advantages

- Relatively few sets of examples are needed to produce a relatively reliable model
- Variance in results is relatively negligible
- Tolerates a low level of noise which can even increase performance
- Evidence based confidence levels helps assess performance

The above points address the disadvantage identified in section 4.8 (insufficient number of examples implies variable result).

Further, EDMs performance with noise eliminates the effect of BER, identified in sections 2.7, 3.9 and 4.8.

Disadvantages

- As problem complexity increases, the number of examples needed increases.

The above disadvantage simply means that the more complex the model the more examples are needed. EDM will still perform to the same level in complex problems if the modeller produces enough examples.

5.12 Conclusions on chapter

This chapter has explored the parameters of neural networks as a means to implement example-giving in EDM. Experimentation has been used to test the limitations of particular aspects of Neural Networks as a means for implementing EDM.

Experimentation has explored four key parameters to the success of EDM: EDM performance and training set size; EDM performance and training set complexity; the robustness of EDM performance and EDM performance with noise.

Results of experimentation are contained in sections 5.3, 5.5, 5.7 and 5.9 for EDM performance and training set size, EDM performance and training set complexity, the robustness of EDM performance and EDM performance and noise respectively.

Using neural networks, EDM can adequately learn simple problems from 25 examples.

More complex problems, up to 7 classifications, can be learnt with 100 examples in the training set. The literature suggests that neural networks can classify more complex problems at the expense of gathering more data.

The reproducibility of results has been shown to be reliable, exhibiting a small variance in performance over 10 identical simulations.

Further, up to 15% of these examples can be incorrect before the classification accuracy of EDM starts to fall, i.e. EDM can tolerate up to 15% error. In addition a 5% noise level improves classification accuracy.

The experimentation in this chapter therefore partially satisfies the third objective, section 1.4:

"Investigate, develop, test and evaluate the proposed novel approach"

Through the use of experimentation, EDM using neural networks has been developed and investigated by exploring the limitations and strengths of the approach.

For example the training set complexity experiment indicates that using 100 examples the highest classification complexity that yields a satisfactory performance is 7.

Another example is the performance of EDM when noise is introduced to the training set. As figure 5.5 shows EDM performs satisfactorily with up to 15% noise, beyond that performance drops to an unacceptable level.

However, the *test* and *evaluate* aspect has not been covered by this experimentation. In the next chapter EDM is used to model real world spreadsheets and the resulting model is compared with the equivalent spreadsheet model.

This allows objective comparison between EDM and traditional spreadsheet modelling which will both tested EDM and also allowed evaluation of the method in the context of spreadsheet modelling.

6.0 The application of EDM in medicine

6.1 Introduction

This chapter presents discussion of possible applications for EDM and some practical experiments show how 'real world' decision support spreadsheets can be modelled using an EDM.

The work contained in this chapter further satisfies the *test and evaluate* aspect of objective 3, section 1.4:

> *"Investigate, develop, test and evaluate the proposed novel approach"*

with regard to the external validity or generality of the results based on the tightly defined problem domain section 3.1.1.

In this chapter, a real world decision support spreadsheet was modelled using EDM, demonstrating the practical usefulness of EDM in certain conditions.

The aim of this chapter was:

> Determine the usefulness of the novel approach by modelling real-world spreadsheets using EDM and testing and evaluating the resulting model against the equivalent real-world decision support spreadsheet.

6.2 Real world decision support spreadsheets

'Real-world decision support spreadsheets' are those that are found in existence designed to fulfil some real decision support function for a professional or an organisation.

There are countless decision support spreadsheets publicly available on the Internet which could be considered for this chapter. However, EDM is more suited to a specific type of problem.

EDM is more suited to decision support spreadsheets containing 'calculative logic' type problems, see section 3.1.1. Calculative logic problems calculate a conclusion based partially or wholly on logical operators, for example credit risk classification.

Calculative logic problems may be coupled with mathematical calculation too, but the emphasis of calculation must be based on logic.

EDM does not perform well with spreadsheets that are purely mathematical, such as a balance sheet, see section 3.1.1

Medicine uses calculative logic for activities such as drug dosing, anaesthesia risk assessment and a variety of other activities. Medicine makes use of spreadsheet technology to implement a variety of different medical calculations.

6.2.1 Decision Support Spreadsheets in medicine

Butler and Croll (2006) discuss the use of spreadsheets in clinical medicine, the study found that spreadsheets exist for critical medical procedures such as anaesthesia dosing in paediatrics and anaesthesia risk assessment for cardiology patients.

Croll and Butler (2006) identified one particular organisation, the Medical Algorithms project (MedAl), as having a high concentration of medical spreadsheets.

To date MedAl had implemented over 10,000 medical algorithms in 45 medical specialties (MedAl, 2007).

The preferred application for the implementation of algorithms is a spreadsheet, specifically Microsoft Excel. As a result MedAl has hundreds of medical decision support spreadsheets available for download on subjects such as balanced nutrition to field treatment for a snake bite.

A large majority of these decision support spreadsheets use calculative logic coupled with some mathematics to execute the medical algorithm in the spreadsheet.

The aim therefore is to determine how accurately EDM can represent the medical algorithms contained in an example medical decision support spreadsheet. In order to carry out this experiment a medical decision support spreadsheet was taken from the medal.org website. The chosen spreadsheet is the Cardiac Anaesthesia Risk Evaluation (CARE) spreadsheet.

6.3 Cardiac Anaesthesia Risk Evaluation (CARE)

The cardiac anaesthesia risk evaluation is defined as:

"...a simple risk classification system for patients undergoing cardiac surgery. This can rapidly stratify a patient for the probability of morbidity and mortality"

(MedAl, 2007)

CARE takes inputs such as cardiac and medical diseases, urgency of surgery and complexity of surgery.

These inputs are used to calculate the likelihood of morbidity (undesired consequences from anaesthesia), prolonged length of stay (the probability of a longer than normal stay in hospital) and mortality (death from anaesthesia).

6.3.1 The CARE algorithm

The CARE algorithm was created by Dupious and Wang (2001) and is intended to be used by anaesthesiologists to determine risk of mortality, morbidity and a prolonged length of stay in hospital.

The CARE algorithm uses 5 inputs and has 8 classifications.

The inputs for CARE are: Cardiac disease (severity) (A), Number of controlled non-cardiac diseases (B), Number of uncontrolled non-cardiac diseases (B), Cardiac surgery (complexity) (C) and Urgency (emergency treatment or otherwise) (D).

All inputs have multiple states, see table 6.1 for input state details.

Input	State 1	State 2	State 3
Cardiac disease (A)	Stable (A1)	Uncontrolled (A2)	Advanced (end stage) (A3)
Medical disease (B)	None (B1)	1 or more controlled conditions (B2)	1 or more uncontrolled conditions (B3)
Cardiac surgery (C)	Non complex (C1)	Complex (C2)	Last hope (C3)
Urgency of surgery (D)	Non Emergency (D1)	Emergency (D2)	N/A

Table 6.1 CARE input state values

The total number of combinations for the input space is 54, i.e. there are 54 unique combinations of the 4 inputs.

However, if you consider that you may have incomplete data, there are 11^2 (2048) possible combinations of input.

There are 8 classifications that can be drawn from the inputs, see table 6.2 for classification detail

Input/ Classification	Cardiac disease (A)	Medical disease (B)	Cardiac surgery (C)	Urgency of surgery (D)
Risk Class 1	A1 AND	B1 AND	C1	?
Risk Class 2	A1 AND	B2 AND	C1	?
Risk Class 3	(A2 OR	B2 OR	C2)	AND D1
Risk Class 4	(A2 OR	B2 OR	C2)	AND D2
Risk Class 5	((A2 OR A3) OR	B3 OR	C2)	AND D1
Risk Class 6	((A2 OR A3) OR	B3 OR	C2)	AND D2
Risk Class 7	A3 AND	?	C3 AND	D1
Risk Class 8	A3 AND	?	C3 AND	D2

Table 6.2 CARE classification summary

In this table there are instances of "?", where this appears the documentation offers no information on what values should be contained here for these classifications.

Currently this algorithm is implemented in a spreadsheet entitled "Cardiac Anaesthesia Risk Evaluation (CARE)" published by MedAl freely available for download.

However, through experimenting with this spreadsheet it is clear that the user can induce a number of serious errors.

6.3.2 The CARE Spreadsheet

The CARE spreadsheet can give some unusual and unexpected results when certain combinations of input are entered.

This is partly due to the combination of input types incorporated into the design of the spreadsheet and partly due to the programming structure employed to implement the algorithm.

The input and output is separated by colour and position on the spreadsheet. Input which is yellow and dark blue is at the top of the spreadsheet and output which is light blue, is at the bottom of the spreadsheet.

6.3.2.1 Errors arising from poor data validation

Some attempt is made to validate the spreadsheet input, these are: Overall data evaluation of input and Validation of input value.

The only example of validation of input value is a cell that instructs the user to input "Y" or "N" if the input is otherwise.

Examples of overall data validation are "data completeness" and "evaluation appropriateness".

The "data completeness" test evaluates if the all input cells contain information, if all cells contain some information the result is "Yes" otherwise it is "No". If the output is "No", the spreadsheet does not calculate any output.

The "evaluation appropriate" test evaluates if the use of the spreadsheet is *appropriate*, i.e. fit for purpose, the output is either Yes or No.

This is based solely on one cell which asks the user if they are evaluating a cardiac patient, input for this cell is either "Y" or "N". If "N" is entered the spreadsheet does not calculate any output.

However, all validation in this spreadsheet is of limited value. All validation checks use the "ISBLANK" function in Excel. The ISBLANK function tests if there is any input in a cell, regardless of what it is.

This means that as long as there is something in the cell, the model still calculates a result even though there has been an input error.

For example, the spreadsheet asks "is surgery an emergency?" the cell requires either "Y,y" or "N,n". If one puts "yes" or "no", which is a feasible mistake to make, the spreadsheet still calculates a result. The calculated result is an error and results in an underestimated risk to the patient.

6.3.2.2 Errors arising from input

Integer values are taken for the following inputs: Number of controlled medical conditions (B2) and Number of uncontrolled medical conditions (B3). There is no minimum or maximum validation placed on the cells.

Although this is not an error, from a programming point of view one should not allow certain values in the cell considering the context. For example the user can enter 1.5 or 20,000 if they wish. Obviously a patient cannot have one and a half controlled or uncontrolled cardiac conditions nor are they likely to have 20,000 simultaneous conditions.

Further because of the ISBLANK function used to validate the cell, one can put non-numerical input in such as the word "three" which considering the question is not beyond the realms of possibility.

Another type of input gathered in the spreadsheet is from a Likert scale. The Likert scale input is taken by using adjacent cells to mark out the possible answers, with the user entering an "X" under the most appropriate cell. See figure 6.1 for an example.

	stable	uncontrolled	end-stage
cardiac disease	X		
	noncomplex	complex	last hope
cardiac surgery	X		

Figure 6.1 Likert scale input (CARE spreadsheet)

Again this is validated using the ISBLANK function, so the user can put anything in the cells and not show an error. This is less serious than other errors since if the user accidentally puts a 1 instead of an X, the calculation is still the same.

However, the user can also enter non-printable characters into the cell such as a space. This can result in cells that don't appear to have any data in because the character is non printable.

6.3.2.3 Errors arising from programming structure

Errors arising from programming structure offer some of the most serious errors in this spreadsheet. The structure causes the spreadsheet to output erroneous risk classifications that in some cases severely underestimate the morbidity, mortality and prolonged length of stay for a patient.

For example, entering input as follows results in category 8 (the highest risk): A3 (End stage cardiac disease); C3 (Last hope cardiac surgery) and D2 (Emergency status). This gives the patient a morbidity, prolonged length of stay and mortality probability of 88.7%, 63.6% and 46.2% respectively.

Notice that in the above scenario, B variables (controlled or uncontrolled medical conditions) are not considered. If any value is entered for uncontrolled medical conditions coupled with the same input that places the patient in risk class 8, the spreadsheet calculates the risk as class 4.

Class 4 risk indicates that the patients probability of morbidity, prolonged length of stay and mortality is 32.1%, 14.7% and 4.5% respectively. This means that with a case more severe than the highest risk class, the probability of morbidity, prolonged length of stay and mortality drops by 56.6%, 42.3% and 41.7% respectively. This is obviously a serious oversight.

The reason why this error and others like it arise is because of the structure of arguments in the spreadsheet. Classes 1,2,3, 5 and 7 are evaluated in one cell. Classes 4 and 8 are evaluated in another cell.

Classes 4 and 8 are arrived at by variable D2 (Emergency surgery), if D2 is true the formula adds one onto the result of the other cell calculating class. For example if the

formula has the conditions that satisfy class 7 and it is an emergency, +1 is added to the class, changing it from 7 to 8.

Because the spreadsheet is coded to look for the specific arguments that match up a class using AND OR and NOT, any unmatchable input using the appropriate logic causes the formula default to the nearest logically true formula.

So when the spreadsheet examines the inputs A3, C3 and D2 the result is class 8, however when examining the inputs A3, B3, C3 and D2 this does not match the conditions of class 8. The formula states class 8 has the conditions A3 AND C3 AND D2.

Specifically the inclusion of B3 therefore fails the class 8 rule, the only other rule that can be true with these inputs is class 4. Class 4 states (A2 OR A3 OR B3) AND C2 AND D2. The inputs (A3 and B3) satisfy the OR conditions and inputs (C2 and D2) satisfy the AND conditions. Therefore the rule is true, therefore the class is set to 4.

Although this has not been commented on by a medically trained professional it would seem that with a case with more severe medical conditions (inputs) than the worst case, the classification should be the highest risk.

At the root of this problem is the fact that classifications are hard coded – i.e. each classification is given a specification that makes the classification true or false, the model only explicitly covers 8 of the 54 possible combinations.

It may be that some of the 54 combinations are either unlikely or invalid from a medical point of view, however it is important to see how the spreadsheet deals with these values since they may cause erroneous output.

The next section deals specifically with unusual combinations of input and records the spreadsheets response.

6.3.2.4 Errors arising from unusual input

Since the spreadsheet relies on hard coded condition checking to classify the input, there are several combinations of input that can or may induce an invalid result. These combinations were arrived at by examining the formulae in the spreadsheet and by experimenting with the spreadsheet. See table 6.3 for the results of the test.

Inputs combinations				Expected outcome	Actual outcome
A1	B1	C1	D1	Class 1	Risk class 1
A2	B1	C1	D1	1 or 2?	Risk class 3
A3	B1	C1	D1	5,6,7 or 8?	"Check data"
A1	B2	C1	D1	Class 2	Risk class 2
A1	B3	C1	D1	5,6,7 or 8?	Risk Class 3
A1	B1	C2	D1	class 3	Risk Class 3
A1	B1	C3	D1	5,6,7 or 8?	"Check data"
A1	B1	C1	D2	3,4 or 5?	Risk class 1
A2	B3	C3	D1	7 or 8	Risk class 3
A2	B3	C1	D1	5 or 6	Risk class 3
A2	B1	C3	D1	7 or 8	"Check data"
A3	B1	C1	D1	5,6,7 or 8?	"check data"
A3	B3	C1	D1	5,6,7 or 8	Risk Class 3
A1	B3	C3	D1	7 or 8	Risk Class 3

Table 6.3 Results from unusual input test

In this table, the input combinations are presented, then the 'expected outcome' and then the actual outcome recorded from the spreadsheet.

In the expected outcome column of table 6.3, some of the expected outcomes have a question mark after them. In these circumstances the expected outcome is an estimate given the severity of input. These have not been checked by a medical professional but are logical given the inputs. Where there is no question mark, the expected outcome is known.

The actual output is colour coded to show that either the output was correct (green), possibly correct (yellow) or definitely incorrect (red).

The results show the spreadsheet incorrectly classifies 10 examples, correctly classifies 4 examples and possibly correctly classifies 1 example.

Of the ten incorrect classifications, four input combinations result in the spreadsheet outputting "check data" (a data validation message). i.e. the spreadsheet failed to give a classification even though it was presented with valid input data.

The other six incorrect examples do provide a classification but according to the documentation concerning the algorithm, the classifications are erroneous.

The one possibly correct classification outputs risk class 3, where the expected output was class 1 or 2, since this is so close and the expected outcome is an estimate, this example is possibly be correct.

6.3.3 Conclusions on CARE spreadsheet

The conclusions drawn from the analysis conducted on the care spreadsheet are as follows:

1. The spreadsheet has multiple series errors and examples of poor spreadsheet programming and design.
2. The CARE spreadsheet has difficulty classifying combinations of inputs that are abnormal, see table 6.3
3. The CARE spreadsheet has been programmed to explicitly test 8 of 54 combinations of input.

In order to make a direct comparison between the CARE spreadsheet and EDM, the CARE algorithm was implemented using EDM and results were compared against those obtained from the CARE spreadsheet.

6.4 Modelling the CARE algorithm with EDM

The aim of the following experiment was to model the CARE algorithm using EDM to determine if doing so eliminated some of the errors and inconsistencies that modelling the problem in a spreadsheet creates.

6.4.1 Aim

The aim of this experiment was:

1. To determine if modelling the CARE algorithm in EDM shows significant advantage over the real world decision support spreadsheet equivalent

6.4.2 Generating the training sets

The training set was generated from the available documentation supplied by MedAl, this is summarised in tables 6.1 and 6.2.

Initially, the training set covered only the cases listed in the documentation. For example, there was only one possible combination, see table 6.2, of classes 1 and 2.

However, class 3 can have three combinations of input, an example of each combination was included in the training set.

Using only the cases in the documentation allows a fairer test between the spreadsheet and the EDM model.

In this example, the training set was a "binary" training set. A "binary" training set is the representation of the original data in a binary format. See table 6.4 for an excerpt of the training set.

A1	A2	A3	B1	B2	B3	C1	C2	C3	D1	D2	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7	Class_8
1	0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0

Table 6.4 Excerpt of EDM training set for CARE algorithm

6.4.3 Neural network selection and performance indicators

The details of the neural network to be used in all experiments in this chapter was discussed in section 4.7

Further, performance was measured in accordance with section 4.6.7.4

6.4.4 EDM CARE algorithm learning results

The results of the EDM CARE model are contained in confusion matrixes tables 6.5 and 6.6. Data contained in table 6.5, showing confusion matrix results for the training set, indicates that the model has been learnt well, the only cause for concern is the 66% 33% split for class 3.

T								
Class	1	2	3	4	5	6	7	8
1	100	0	0	0	0	0	0	0
2	0	100	0	0	0	0	0	0
3	0	33.3	66.6	0	0	0	0	0
4	0	0	0	100	0	0	0	0
5	0	0	0	0	100	0	0	0
6	0	0	0	0	0	100	0	0
7	0	0	0	0	0	0	100	0
8	0	0	0	0	0	0	0	100

Table 6.5 Confusion matrix T value results

Results in table 6.6, which were generated using the cross validation set, are notably different to those contained in table 6.5. The results in table 6.6 indicate problems were encountered during the learning process. For example classes 1 and 2 both have

50% splits, i.e. accuracy is split between the right class and the wrong class in equal proportions for both classes 1 and 2.

CV								
Class	1	2	3	4	5	6	7	8
1	50	0	50	0	0	0	0	0
2	0	50	50	0	0	0	0	0
3	0	33.3	66.6	0	0	0	0	0
4	0	0	0	100	0	0	0	0
5	0	0	0	100	0	0	0	0
6	0	0	0	0	0	16	84	0
7	0	0	0	0	0	0	88	11
8	0	0	0	0	0	0	88	11

Table 6.6 Confusion matrix CV value results

Tables 6.5 and 6.6, the training and cross validation confusion matrixes, show significant difference.

The positive result in table 6.5 shows that the model has learnt the problem well based upon the training set only, i.e. considering only the training data EDM has performed well.

The less positive result contained in table 6.6 suggests that the model may have difficulty generalising to unseen examples.

Both tables must be considered when evaluating results, a difference in performance between the training and cross validation sets as seen in tables 6.5 and 6.6 suggests a difficulty extrapolating to unseen examples.

However, the best measure of performance is testing and in order to fairly compare the EDM CARE model and the CARE spreadsheet, the same testing must be applied to the EDM CARE model as it was to the CARE spreadsheet.

6.4.5 EDM performance with unusual input

To make a direct comparison between the EDM CARE model and the CARE spreadsheet, the same 'unusual input', see table 6.3, was passed through the EDM CARE model. The results of this are contained in table 6.7.

Inputs				Expected	CARE spreadsheet	EDM model
A1	B1	C1	D1	Class 1	Risk class 1	Class 1
A2	B1	C1	D1	1 or 2	Risk class 3	Class 1
A3	B1	C1	D1	5,6,7 or 8	"Check data"	class 6
A1	B2	C1	D1	Class 2	Risk class 2	class 2
A1	B3	C1	D1	5,6,7 or 8	Risk Class 3	class 5
A1	B1	C2	D1	class 3	Risk Class 3	class 3
A1	B1	C3	D1	5,6,7 or 8	"Check data"	class 3
A1	B1	C1	D2	3,4 or 5	Risk class 1	class 1
A2	B3	C3	D1	7 or 8	Risk class 3	class 8
A2	B3	C1	D1	5 or 6	Risk class 3	class 1
A2	B1	C3	D1	7 or 8	"Check data"	class 6
A3	B1	C1	D1	5,6,7 or 8	"check data"	class 6
A3	B3	C1	D1	5,6,7 or 8	Risk Class 3	class 6
A1	B3	C3	D1	7 or 8	Risk Class 3	class 3

Table 6.7 EDM and spreadsheet performance with unusual input

Table 6.7 shows the EDM model of the CARE algorithm has more success with abnormal input than the CARE spreadsheet. The same colour coding is used in table 6.7 as it the original test, table 6.3.

As can be seen in table 6.7, the EDM model of the CARE spreadsheet outperforms the CARE spreadsheet in the abnormal input test. The CARE EDM model classifies 9 of the 14 examples correctly comparatively the CARE spreadsheet classifies 3 of the 14 examples correctly.

The EDM model is therefore better at dealing with unusual input than the equivalent CARE spreadsheet. The reason that the EDM CARE model outperforms the CARE spreadsheet is because of a property of Neural Networks.

Neural networks possess the property 'gradual degradation' which means in circumstances where the network is being pushed to breaking point, failure is gradual rather than abrupt.

Since the CARE spreadsheet is based on Excel programming, failure is more abrupt than the gradual degradation observed in neural networks.

6.4.6 Blind testing sets for CARE spreadsheet and EDM model

The testing contained in table 6.7 shows that the performance of EDM is superior when considering 'unusual' input. However, the testing set in this instance is designed to contain extreme values to test the boundary of the systems. Therefore a randomly selected blind test was also conducted.

The total possible number of combinations for the CARE algorithm is 54, i.e. there are only 54 possible combinations of input and output. The Blind testing set was therefore 54 for both the EDM CARE model and the CARE spreadsheet.

The EDM CARE model misclassified 4 of the 54 examples giving it a classification accuracy of 93%. The CARE spreadsheet misclassified 11 of the 54 examples giving it a classification accuracy of 80%.

6.5 Conclusions on modelling the CARE algorithm with EDM

The manner in which the CARE spreadsheet deals with unusual input highlights the advantages of using neural networks to implement EDM and advantage of using EDM over using spreadsheets for the CARE model. Gradual degradation prevents the EDM model from abrupt failure, unlike the CARE spreadsheet.

This contrast is described in figure 6.2 which shows the difference in performance degradation between the EDM CARE model and the CARE spreadsheet.

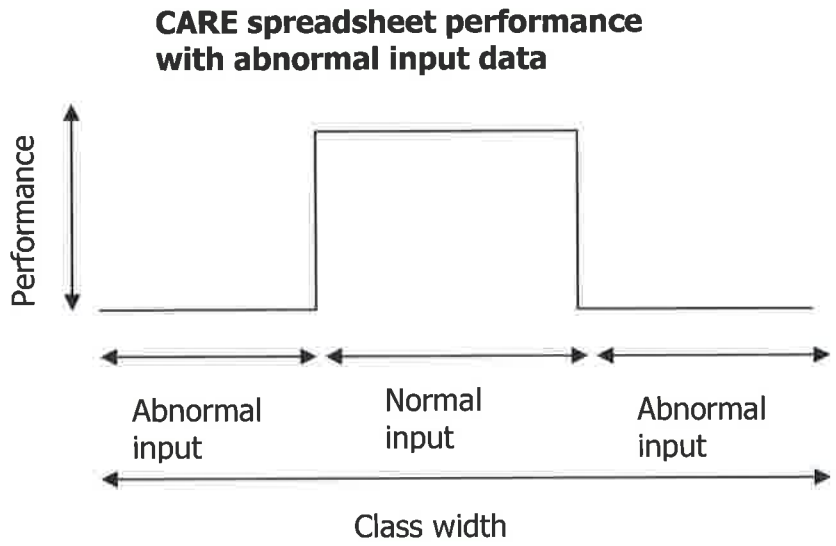

Figure 6.2 CARE spreadsheet performance with normal and abnormal input data

As can be seen in figure 6.2, the CARE spreadsheet performs as it should with 'normal' or expected input. Once the input is outside what is expected, there is a sharp drop in performance.

The EDM model performs with abnormal input by gradual degradation in performance, figure 6.3 describes this gradual degradation.

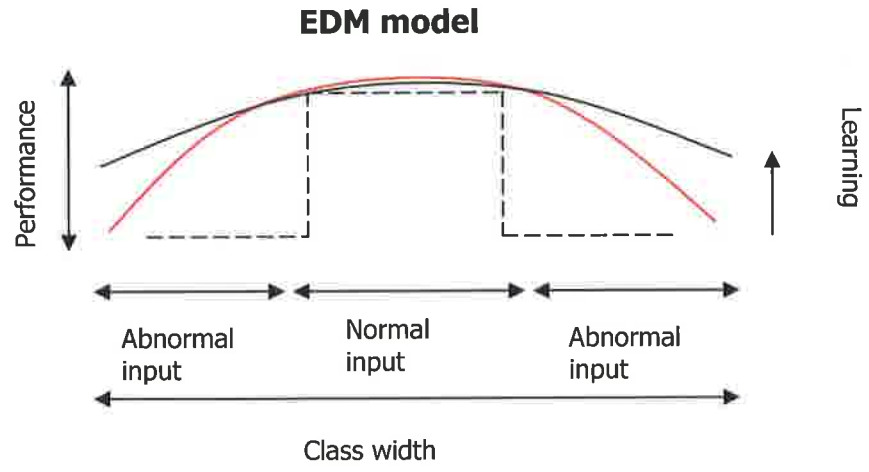

Figure 6.3 EDM CARE model performance with abnormal input data

As can be seen in figure 6.3 the degradation in performance, shown for EDM as a curved line, is a gentler gradient than that of the CARE spreadsheet. For reference the CARE spreadsheet performance is included in figure 6.3 as a dotted line.

Further, learning and performance in the EDM model can be improved by providing more examples to learn from and more time to learn in. This is represented in figure 6.3 by the additional curved line above the red line.

However, EDM cannot be applied universally, i.e. EDM only works in certain types of spreadsheet model.

6.6 Limitations and strengths of EDM application

Although EDM performs well in the CARE algorithm model above, it is not suited to some types of spreadsheet model.

Spreadsheet models that are only numerical cannot be modelled using EDM, i.e. a spreadsheet that consists solely of mathematical calculations cannot be modelled using EDM.

EDM works well in spreadsheet models that make use of logic and mathematical calculation. In spreadsheets where logic is not present, EDM does not work. Spreadsheet models that use mathematics and logic are often 'decision support systems' that recommend some course of action based upon a number of numerical inputs and logical operations.

The CARE spreadsheet is a prime example of the type of spreadsheet EDM is suited to. The CARE spreadsheet combines some mathematical values with logic operators to give an assessment of patient mortality, morbidity and prolonged length of stay risk. These risks are bound to broad categories which the CARE spreadsheet indicates are true if the priori conditions are such.

Further, examples exist outside of medicine, in credit risk analysis similar models exist that place a participant in a number of categories based upon mathematical values and logic tests, see figure 6.4.

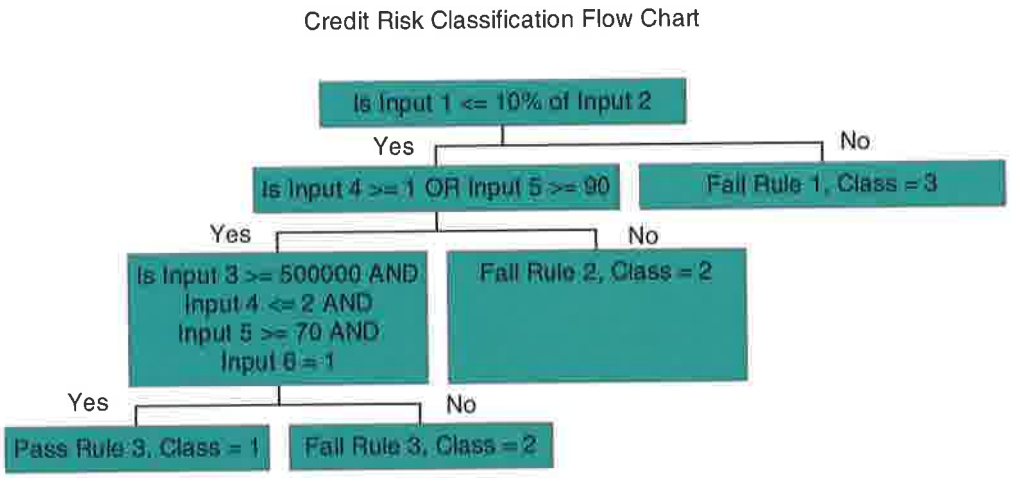

Figure 6.4 Example credit risk classification model

The exact number or proportion of spreadsheets that use a mathematics and logic combination is unclear and has never been explicitly investigated, however studies into the use of functions can indicate how common mathematical and logic operators are.

6.6.1 Functional operators in spreadsheets

As defined by the vendor Microsoft, there are 11 classes of function offered with the standard Excel spreadsheet software. Excel is chosen since it is the most commonly used spreadsheet application according to Walchenbach (2005). Walchenbach states that Excel now accounts for 90% of the spreadsheet market, although it is difficult to determine the exact number of Excel users, in 1997 alone Microsoft shipped over 70 million copies of Excel 97.

These classes contain operators to be used in formulae expressions and are grouped according to their actual purpose. The 11 classes contain varying amounts of operators

ranging from 5 to 78 operators in a class, offering a total of 343 unique operators. The 11 class groupings are shown in table 6.8.

Class Name	Number of operators
Database	12
Date and Time	20
Financial	53
Engineering	39
Information	18
Logical	6
Look-up and Reference	17
Math and Trigonometry	60
Statistical	78
Text	35
External linking	5

Table 6.8 Excel function classes

6.6.2 The use of functions in spreadsheets

As previously mentioned, there have been no studies identifying the number of spreadsheets using mathematics and logic in the same manner as the CARE spreadsheet, however there are studies focusing on which functions are used commonly in spreadsheets.

Chan and Storey (1996) surveyed 256 analysts using Lotus 123 on the functionality of spreadsheets used, the results of which are shown in figure 6.5.

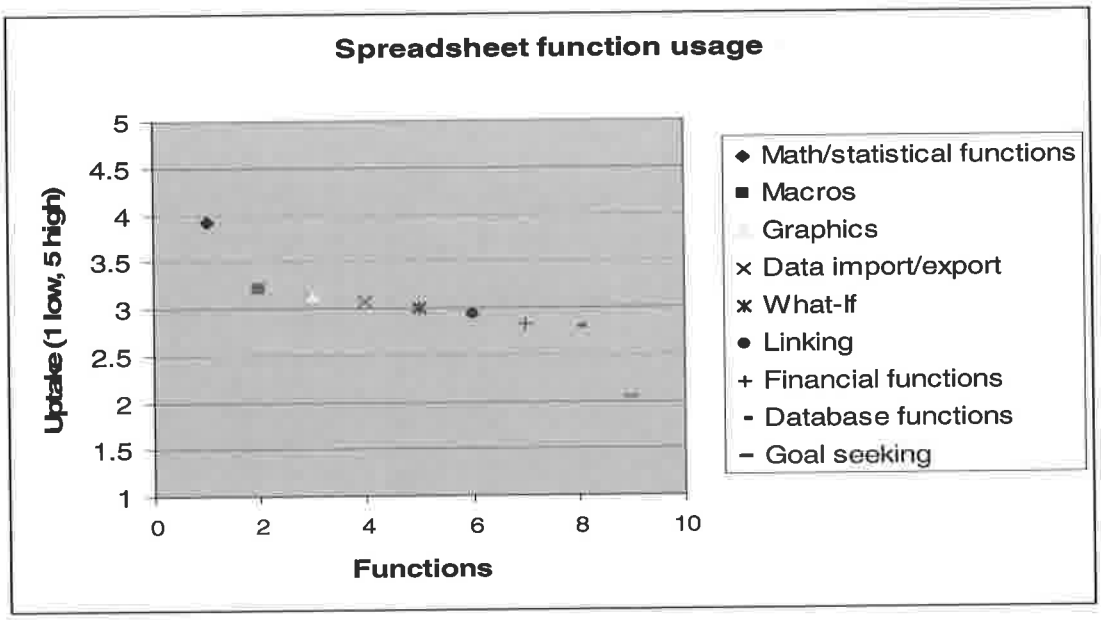

Figure 6.5 Chan and Storey (1996)

Figure 6.5 shows Likert scale results, the participants indicated how often they use a particular class of function in their spreadsheet and that was recorded on a Likert scale (1 being never and 5 being All the time).

Figure 6.5 shows that mathematical and statistical functions are the most frequently used and that goal seeking is the least used. However, since this study was conducted on Lotus 123 users, the functional classes are different to those of Excel.

Unfortunately, the vendor Lotus was unable to provide a detailed functionality listing for Lotus 1-2-3. This difference makes direct comparison difficult which is exacerbated since some operators in Excel are not supported in Lotus 1-2-3 and visa-versa.

Ballinger *et al.* (2003) presented data collected and from 259 workbooks used to record student marks in a University. Figure 6.6 shows the results of the survey, in this case the results are expressed as a cumulative frequency of class operators rather than a proportion, i.e. the number of individual logic operators used in the 259 workbooks.

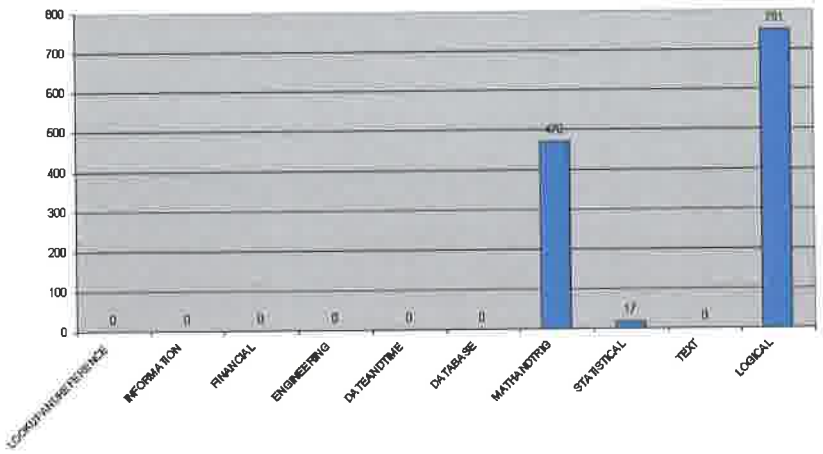

Figure 6.6 (Ballinger *et al.*, 2003)

Figure 6.6 shows that logical and mathematical functions are used more extensively than other classes. However, this seems disproportionate considering figure 6.5. Since the sample was taken from a university and the spreadsheets were used to record student marks, it is possible the sample is biased, i.e. to record student marks very few classes of operators are needed.

On the other hand figure 6.6 does reflect figure 6.5 to some extent, both studies identify mathematical functions are used extensively. However, it is unclear if Chan and Storey (1996) include the Logic operators in their mathematics class.

The most comprehensive and up to date survey of spreadsheet functionality usage is part of a study conducted by the Spreadsheet Engineering Research Project (SERP) at the Tuck School of Management, Dartmouth.

SERP (2006) presents functionality survey results as part of a wider spreadsheet usage study. The study took 35 randomly selected spreadsheets submitted by the school's alumni. The spreadsheets were then audited and information, including functional operator usage, was extracted.

Figure 6.7 shows the results from the SERP (2006) survey. In this survey the functional operators are expressed as a percentage of the total. In other words the figures for each of the functional classes in figure 6.7 are expressed as a percentage of the whole sample.

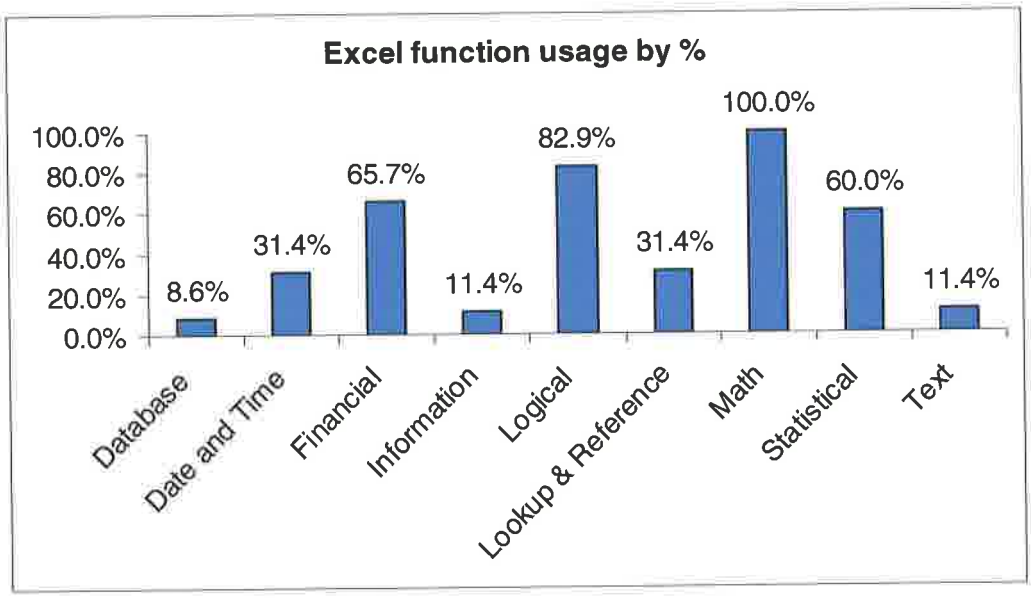

Figure 6.7 (SERP, 2006)

Figure 6.7 shows that mathematical and logical functions are used the most followed by financial and statistical. More specifically, mathematical functions feature in 100% of the spreadsheets audited and logical functions feature in 82.9% of the spreadsheets audited.

All three studies (SERP 2006, Ballinger 2003 and Chan and Storey 1996) identify that Math functions are used extensively in spreadsheets. Both SERP (2006) and Ballinger (2003) identify that Logic and mathematical functions are used extensively.

If we consider the possibility that Chan and Storey interpret logical operators as part of the math class, the Chan and Storey survey also indicates that spreadsheets are predominately made up of math and logic classes.

6.7 Conclusions on the use of functions in spreadsheets

We conclude that spreadsheets are predominately made up of mathematical and logic operators. This is based on the evidence presented by SERP (2006), Ballinger *et al.* (2003) and to some extent Chan and Storey (1996).

Of the three studies, SERP (2006) is the most recent and comprehensive, so it is likely that SERP (2006) offers a more accurate reflection of the current situation regarding the composition of spreadsheets. A good summary of this work is contained in Thorne and Ball (2006a) and work considering the use of functional operators in contained in Thorne and Ball (2006b)

EDM has several practical applications in several industries. As shown in this chapter, EDM could be used to model risk and diagnosis in medicine, especially considering the CARE medical spreadsheet in this chapter has been shown to be error prone.

In broader terms, EDM is applicable to decision support spreadsheets such as the CARE spreadsheet model or the credit risk classification model as shown in figure 6.4.

Further, since it is the type of problem that is critical to the application of EDM, i.e. decision support spreadsheets that use logic and mathematics, EDM has applications outside of spreadsheets since spreadsheets are not the only tool used to implement decision support systems.

6.8 The applicability of EDM to the spreadsheet error problem

As with all new thinking and novel ideas, EDM has a specific application to a relatively small number of spreadsheet errors, i.e. EDM is not a panacea.

As shown in figure 6.8, it is likely that EDM *solves* a small percentage of spreadsheet errors that exist in the spreadsheet problem.

Other research areas will solve other aspects of the spreadsheet error problem, for example spreadsheet engineering or test driven development in spreadsheets will solve certain small, but significant, aspects of the spreadsheet error problem.

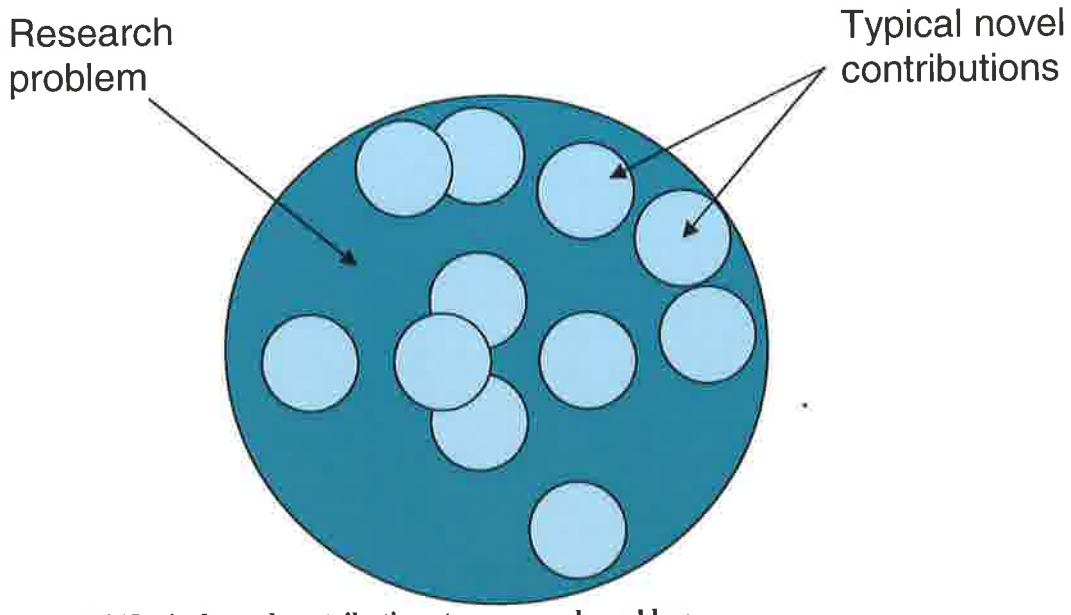

Figure 6.8 Typical novel contributions to a research problem

6.9 Advantages and disadvantages gained when applying EDM

This section summarises the main advantages and disadvantages gained when applying EDM to real world decision support spreadsheet problems.

Advantages

- Gradual degradation prevents sudden or rapid failure
- More accurate than CARE spreadsheet, especially considering extreme parameter values

The above advantages are further to those identified in sections 3.9, 4.8 and 5.11

Disadvantages

- EDM can only be applied to spreadsheets using a combinations of mathematics and logic or purely logic (decision support spreadsheets)

6.10 Conclusions on the chapter

The aim for this chapter was, section 6.1:

> *Determine the usefulness of the novel approach by modelling real-world spreadsheets using EDM and testing and evaluating the resulting model against the equivalent real-world decision support spreadsheet.*

This chapter demonstrates how EDM can be used to model a real world decision support spreadsheet taken from the medicine. Further, EDM has shown itself to be more reliable and consistent than the equivalent real world decision support spreadsheet model, see sections 6.3 and 6.4

The superior performance, especially when considering unusual or extreme input, is largely down to 'gradual degradation' that EDM inherits from using neural networks to implement EDM, see section 6.5.

Gradual degradation gives EDM the ability to produce better results than the equivalent decision support spreadsheet model in adverse conditions because EDM can deal with uncertainty, noise and incomplete input in a more reliable manner than the spreadsheet.

This suggests a novel application for EDM might be useful in overcoming some of the difficulties with omissions of domain, world and semantic knowledge (McDaid and Fitzgerald, 2008) in traditional requirements analysis.

The successful application of EDM to the CARE spreadsheet, or any decision support spreadsheet model, is dependant on the spreadsheet functions used in the original spreadsheet, see section 6.6.

If the decision support spreadsheet uses mathematical and logic functions together, it is likely that EDM can be used to model the problem, for example the CARE spreadsheet or the credit risk analysis model in figure 6.4.

Further, the above examples can be described as Decision Support Systems (DSS), so any DSS combining logic and mathematics could potentially be modelled using EDM. This may mean that EDM can be used to model problems outside of spreadsheets, i.e. EDM could have applications beyond decision support spreadsheets, see section 6.7

However, if the spreadsheet uses only calculative mathematics, EDM cannot be used to model it, see section 3.1.1. Examples of this include balance sheets and profit and loss accounts.

The above conclusions and discussions satisfy the *test* and *evaluate* aspects of third objective of this thesis. In combination with the work in chapter 5, the third objective is now fully satisfied.

"Investigate, develop, test and evaluate the proposed novel approach"

This completes the primary work in this thesis, the next chapter will evaluate and reflect on the thesis as a whole offer limitations, strengths and areas for further work.

7.0 Conclusions, Reflections and Further Work

7.1 Introduction

This chapter identifies the main contribution to knowledge of this thesis by revisiting the various **conclusions** of the primary and secondary work with respect to the research question, aim and objectives. There follows **reflections** on the issues set out in earlier chapters. Lastly **further work** possibilities are discussed.

7.2 Conclusions

The motivation for this thesis was the realisation that spreadsheet errors are both prevalent and have significant impact and there was an opportunity for novel research.

In particular combining spreadsheets with some form of machine learning technique was of particular interest. Potentially machine learning techniques could be used to reduce some of the errors found in spreadsheets. Hence the research question stated in section 1.4.1 (and answered affirmatively in the rest of the thesis) was:

> *Is it possible to create an alternative modelling technique for the reducion of error in decision support spreadsheets?*

And the aim of the thesis as stated in section 1.4.2 (and answered affirmatively in the rest of the thesis) was:

> *To create and evaluate an alternative modelling technique for the reduction of error in decision support spreadsheets*

7.2.1 Revisiting objective 1 (Literature review)

Objective 1 for the thesis was defined in section 1.4.3 as follows:

> *Undertake a literature review of relevant topics within the field of spreadsheet error research*

Spreadsheet error research was introduced in section 2.2

Examples of current spreadsheet error research comprise experiments into spreadsheet error, taxonomies of spreadsheet error, observations of spreadsheet error in practice, theories on spreadsheet error management, manual auditing and auditing software.

Experiments concerning spreadsheet error offer some quantification of the magnitude of spreadsheet error and impact that errors can have on organisations. Although error rates vary, it is clear from the evidence available that at least 30% of spreadsheets contain error (section 2.2).

Taxonomies of error provide a means to classify error. However, inconsistency in terms, discrepancies in error classification and prescriptive structures make these taxonomies problematic to apply.

Panko has argued that spreadsheet errors are complex but similar to human error. It has been suggested by Panko that human and spreadsheet errors are closely related. Moreover, Panko suggests that spreadsheet error rates are approximately the same as

error rates found in other human activities such as spelling, typing or programming a computer.

Human factors play a significant role in spreadsheet errors and have been largely ignored by the wider spreadsheet community. Some authors suggest that spreadsheet errors are human errors found in spreadsheets, see section 2.9.

The quantifiable human factors such as BER and Cognitive load have been shown to have an effect on spreadsheet quality. Other aspects such as Miller's threshold bear on the relationship between spreadsheets and error.

The unquantifiable human factors overconfidence and bias significantly effect spreadsheet model quality. Overconfidence has also proven difficult to mitigate

Chapter 2 concluded in section 2.8 with the opportunities for novel research were:

> *The novel approach would use real world examples provided by the user. i.e. the user would think up examples of input and output for a given problem.*
>
> *The computer, using a machine learning technique, would deduce the mathematics and logic of the examples and generate a 'model' to reflect the examples.*
>
> *It is thought that this approach could reduce spreadsheet error by reducing the impact of certain human factors during development.*
>
> *For example, the cognitive load of thinking up examples should be significantly lower than that of spreadsheet modelling*

7.2.2 Revisiting objective 2 (Develop alternative modelling technique)

Objective 2 for the thesis was defined in section 1.4.3 as follows:

> *Based upon the literature review, consider an alternative modelling technique for the reduction of error in decision support spreadsheets*

Section 3.2 stated concerning chapter 3:

> *The purpose of this chapter is to establish if the novel approach identified in the literature review,* ***"example-giving"*** *(see section 2.8), is feasible.*

The **novel concept** is that humans find it easier to provide examples (attribute classifications) than creating formulae in spreadsheets and that if this technique yields a significant advantage over traditional spreadsheet modelling, then the resulting examples could be used create a less error prone decision support model.

The above statement is also covered by the research question of this thesis, see below.

> *Is it possible to create an alternative modelling technique for the reducion of error in decision support spreadsheets?*

Section 3.7.7 shows the results of an experiment designed to test if example-giving was more a more accurate tool than creating the equivalent decision support spreadsheet model in Excel when considering a number of written exercises.

The results of this experiment show that there is a statistically significant accuracy advantage to be gained from example-giving over traditional decision support spreadsheet modelling in Excel.

Section 3.10 summarised the investigation of the feasibility of example-giving and suggested that example-giving is likely to be more useful in more complex scenarios.

Chapter 3 showed the feasibility of example-giving and chapter 4 discusses and decides upon the approach to implement EDM. The advantages and disadvantages of various approaches to implement example-giving were considered and section 4.3.8 concluded:

> *The chosen approach to implement example-giving is machine learning, this is because in comparison to the other potential approaches it is more automated, is superior at coping with complexity and noisy data sets and is offers a more stable repeatable process than other approaches.*

Further investigations of machine learning algorithms resulted in the decision to use Neural Networks. This was because of the following advantages as discussed at length in section 4.4.5.4:

- The ability of Neural Networks to deal with noise by gradual degradation is great strength.
- Potentially this may allow EDM to be tolerant of user error such as BER
- Neural Networks are mostly self programming and self organising by the user providing examples.
- With EDM in mind, the self programming ability of neural networks may reduce the number of errors that arise from poor programming in spreadsheets.
- The ability to give evidential responses (providing levels of confidence in results) is also cited as a major strength
- The ability to give evidential responses could potentially be used by the EDM modeller to determine the reliability of the EDM model.
- The ability to generalise, although not exclusive to Neural Networks, is another great strength.
- Neural networks have been successfully applied to many similar problems that relate to decision support activities e.g. Bankruptcy prediction, Cardiac disease diagnosis or classification of level of return on stock investments.
- The common factor in the above applications of neural networks is that all of them are classification problems. Neural networks are particularly strong at classification problems.

- Since decision support neural networks have been successfully applied to classification problems, it is reasonable to expect successful application to the decision support activities found in spreadsheets.

Hence as discussed in sections 4.5.3.1 and 4.5.3.2 a convenient Neural Network package with genetic optimisation (see section 4.4.5.3) called "**Neurosolutions**" was adopted for all further EDM experimentation.

7.2.3 Revisiting objective 3 (Primary research)

Objective 3 for the thesis was defined in section 1.4.3 as follows

Investigate, develop, test and evaluate the proposed novel approach

Chapter 5 considered the practical parameters of the novel approach, specifically the number of examples needed, the effect of complexity on performance, the sensitivity of the learning process and the effect of noise on performance. This work in part satisfies objective 3 by establishing some of the performance parameters of the novel approach, i.e. *investigate and develop*. However, it does not deal with the *test and evaluate* aspects of objective 3 which are addressed in chapter 6.

Section 5.11 discussed the advantages and disadvantages of EDM implemented with Neural Networks as:

Advantages

- *Relatively few sets of examples are needed to produce a relatively reliable model*
- *Variance in results is relatively negligible*
- *Tolerates a low level of noise which can even increase performance*
- *Evidence based confidence levels helps assess performance*

The above points address the disadvantage identified in section 4.8 (insufficient number of examples implies variable result).

Further, EDMs performance with noise eliminates the effect of BER, identified in sections 2.7, 3.9 and 4.8.

Disadvantages

- *As problem complexity increases, the number of examples needed increases.*

The above disadvantage simply means that the more complex the model the more examples are needed. EDM will still perform to the same level in complex problems if the modeller produces enough examples.

Chapter 6 examines the practical performance of the novel approach using 'real world' decision support spreadsheet (the CARE spreadsheet model). The novel approach is tested and evaluated against the CARE spreadshee model and comparisons drawn. This in turn allowed an evaluation of the novel approach when compared a real world decision support spreadsheet.

Section 6.4.5 discussed the CARE EDM model and the CARE spreadsheet and demonstrated the EDM model to more reliable and accurate than the CARE spreadsheet.

The reason for this greater reliability and performance is gradual degradation that is inherited by EDM from using neural networks as the means of implementation, see figure 6.3.

Section 6.10 summarises chapter 6 and emphesises that the superior performance of EDM is largely down to '*gradual degredation'*, especially dealing with uncertainty, noise and incomplete input.

7.3 Reflections

7.3.1 Origins of example-giving and EDM

Section 2.7 discussed spreadsheet errors as a mismatch between man and machine. Section 2.5.4 stated that, human factors play a significant role in spreadsheet errors and have been largely ignored by the wider spreadsheet community.

In section 2.7 we discussed the origins of the novel contributions of this thesis. We stated that Michie had argued that human computer interaction was fundamentally limited due to the way in which humans interact with the computer. Michie essentially pointed out that the roles of machine and human in interaction did not exploit either's strengths. In that vein, we argued that spreadsheet errors are mainly attributed poor interaction between humans and computers.

Hence a potentially more beneficial paradigm would be to play on the natural strengths of the human and the conventional computer. In this new paradigm, the human would pattern match and generate real world examples, the computer would use its ability of mathematical manipulation and logical deduction to build a model from the examples provided by the user, see the circled sections on table 2.8 (reproduced from chapter 2).

	Pattern matching	Generating real-world examples	Manipulating mathematics	Logical deduction
Human	Strong	Strong	?	?
Conventional Computer	Weak	Weak	Strong	Strong

Table 2.8 Proposed methods of interaction

The example-giving technique in EDM allows the human to provide examples and pattern matches whilst the machine computes the mathematics and logic which plays on the strength of both, see table 2.8 above. In this vein the impact of human factors is significantly reduced when using EDM.

In conclusion reflecting on these ideas, this seems to be very solid ground although the resulting EDM solution is not necessarily the only way of implementing these ideas which could be explored in further work.

7.3.2 Discussion of the wider issues of EDM

7.3.2.1 Advantages of example-giving in EDM

Section 3.9 stated the following:

Advantages of example-giving:

- *Example-giving is easy (see section 3.6.4)*
- *Eliminates the need to program the computer*
- *Eliminates BER in the programming of a spreadsheet*
- *Eliminates bias in the spreadsheet*

From the above advantages of example-giving, it would seem to be that many sources of the problem of spreadsheet error (section 2.7) such as BER, Poor programming and bias are reduced or eliminated.

7.3.2.2 Disadvantages of example-giving in EDM

Section 3.9 stated the following:

Disadvantages of example-giving:

- *May introduce some BER in the proposed alternative method, i.e. creation of examples*
- *May introduce bias in the creation of examples*

7.3.2.3 Limitations of EDM

Since EDM makes extensive use of Neural Networks, the criticisms that apply to NN also in part apply to EDM. Some of these criticisms were answered by making use of Genetic Optimisation in conjunction with Neural Networks.

For example by genetically optimising the input space of the Neural Networks, one can relieve the issues of small training sets, see section 4.4.5.4

Unfortunately, nothing can be done to alleviate the 'black-box' syndrome that Neural Networks exhibit. The 'black-box' syndrome is accepted as a necessary cost to using Neural Networks. However this is a criticism of Neural Networks not necessarily of EDM since the examples will be understandable to a domain expert.

EDM can learn relatively simple problems with as few as 25 examples, see section 5.3. More complex problems require a greater number of examples, however not excessively so, see section 5.5.

The quality and coverage of examples is important because if the examples provided by the user do not cover all of the classifications in the model accurately, the resulting EDM model reflects those inaccuracies, i.e. rubbish in rubbish out.

Hence the performance of an EDM model is based upon the number of examples provided, the quality and coverage of those examples, the complexity of the examples, the amount of noise present and the amount of time available.

The level of noise present affects the ability of EDM to learn the problem, see section 5.9. Where the level of noise is 15% and below, the performance of EDM is actually increased. When this noise level exceeds 15%, EDM learning is worsened, see figure 5.3. An example of noise could be BER in the generation of examples for the EDM model.

Learning time can be under five minutes but for more complex problems with large data sets, learning can take over an hour. Further, the amount of time available to learn the problem may also affect the accuracy of the resulting EDM model.

Essentially, the more time available (typically an hour or more) the better the resulting model, this is also true for the number of examples, i.e. the greater the number of examples the better the resulting EDM model.

Users may feel unfamiliar with the EDM and the example-giving method that in practice may make EDM difficult to use. However, results gathered by the experimentation in chapter 3 suggests the EDM users (treatment group) found using EDM '*easier*' than the spreadsheet modelling group using Excel, see figure 3.14.

Spreadsheet models in the financial industry for example are often used merely to decide between alternatives rather than to generate precise figures. However, spreadsheet users may find the EDM approach of percentage accuracy rather than a complete accuracy to be a psychological hurdle.

7.4 Further work

7.4.1 Full trial of EDM

The most obvious extension to the work contained in this thesis would be a full trail using participants from industry to fully understand the practical strengths and limitations of EDM.

First, this would require the development and integration of neural networks package Neurosolutions into a software package such as Excel.

Second, the industry staff involved would need to have the example-giving thoroughly explained to them and some trialled use of the Neurosolutions package.

Third, the EDM package ought then to be compared by parallel trialling against the traditional spreadsheet modelling technique.

Preferably these trials should be done with sufficient individuals to be statistically significant and in more than one domain.

7.4.2 Requirements analysis

The process of giving-examples, i.e. thinking about the whole problem and giving examples that cover the entire specification, suggests a further use of EDM. The graceful degradation of EDM does seem to have some advantages in detecting errors of omission, since the process of 'thinking up' examples to be used for training makes the modeller consider the *whole* problem.

This provides a unique handle on one of the trickiest problems in traditional systems analysis, i.e. getting around the problem of semantic omissions, world knowledge omissions and domain omissions. Trials on this basis should be attempted to test whether this suggestion is indeed correct which would then be very advantageous since there are virtually no other techniques that correct these problems.

7.4.3 Test Driven Development

Another possibility for further work might be to explore the extent to which Test Driven Development (TDD) tests could be considered as examples for EDM training. The reason for considering combining EDM with TDD is that TDD tests closely resemble training sets in EDM, it would seem that some beneficial combination is possible.

As discussed in section 4.3.6, Test Driven Development (TDD) is an agile method for developing specifications and code for software products. In TDD the developer writes test cases first and then writes code that attempts to satisfy each test. If the code satisfies the test, the developer moves on to the next test case otherwise the developer

modifies the code until it satisfies the test. This process is repeated until the developer cannot think of any other test cases.

For example in order to write good TDD test cases, the developer must consider the input and corresponding output of the model, this has considerable overlap with EDM. In TDD the developer considers the input and output of the **software,** in EDM the modeller thinks of the attribute classifications of the **problem,** which is a higher level. This has considerable similarity with EDM which suggest possibilities for further work.

For example, EDM could be used to check the validity and completeness of TDD tests before the test are used to write code. This would bring the possible benefit of broader thinking to the TDD cases.

References

Aamodt. A, (1994), *'Case Based Reasoning: Foundational issues, Methodological variations and systems approaches'*, AI communications, 7 (1), pp 39-59

Abbass. H, (2002), *'An evolutionary artificial neural network approach for breast cancer diagnosis'*, Artificial Intelligence in Medicine, 25 (3), pp 265-281

Akiyama, F., (1971), *'An Example of Software Debugging'*, Paper presented at the Proceedings of the International Federation of Information Processing Societies (IFIPS).

Alavi. M, Nelson.R, Weiss, (1987), *'Strategies for End User Computing: An integrative framework'*, Journal of Management Information systems, 4 (3), pp 28-50

Alavi.M and Weiss.I, (1985), *'Managing the risks associated with End User Computing'*, Journal of Management Information Systems, 2, (3), pp 16-21

Allwood. C, (1984), *'Error detection processes in statistical problem solving'*, Cognitive Science, 8, pp 413-437

Ambler. S. W, (2008), *'Introduction to Test Driven Development'*, Internet http://www.agiledata.org/essays/tdd.html, Status: Available, accessed 3.08.08 12.49pm

Andreou. P, Martzoukos. S, Charalambous. C, (2008) *'Pricing and trading European options by combining ANNs and parametric models with implied parameters'*, European Journal of Operational Research, 185, pp 1415-1433

Armor. D and Taylor. S, (2000), *'Mindset, prediction and performance: self regulation in deliberative and implemental frames of mind'*, Not published, Yale University

Atiya, A. F., (2001), *'Bankruptcy prediction for credit risk using neural networks: A survey and new results'*, IEEE Transactions on Neural Networks, 12(4), pp 123-149

Ayalew, Y., M. Clermont, R. Mittermeir. (2000), *'Detecting Errors in Spreadsheets'*, Proceedings of EuSpRIG Symposium, EuSpRIG 2000 Symposium, Spreadsheet Risks—the Hidden Corporate Gamble. (pp. 65-76). Greenwich, England: Greenwich University Press.

Azuaje, F. Dubitzky, W. Wu, X. Lopes, P. Black, N. Adamson, K. White, J.A., (1997), *'A neural network approach to coronary heart disease risk assessment based on short-term measurement of RR intervals'*, Computers in Cardiology, pp 53-56, NIBEC, Ulster University

Baddeley, A. D., & Longman, D. J. A. (1978), *'The Influence of Length and Frequency of Training Session on the Rate of Learning to Type'*, Ergonomics, 21(8), 627-635.

Baesens. B, Viaene. S, Van den Poel. D, Vanthienen. V, (2002), *'Bayesian neural network learning for repeat purchase modelling in direct marketing'*, European Journal of Operational Research, 138, (1), pp191-211

Ballinger. D ,Biddle. R ,Noble. P, (2003), *'Spreadsheet structure inspection using low level access and visualisation'*, Proceedings of the Fourth Australian user interface conference on User interfaces 2003, p.91-94, February 01, 2003, Adelaide, Australia

Bandura, A. (1994). *'Self-efficacy'*. In V. S. Ramachaudran (Ed.), *Encyclopedia of human behavior* (Vol. 4, pp. 71-81). New York: Academic Press. (Reprinted in H. Friedman [Ed.], *Encyclopedia of mental health.* San Diego: Academic Press, 1998).

Basili, V. R., & Selby, R. W., Jr. (1986), *'Four Applications of a Software Data Collection and Analysis Methodology'*, In J. K. Skwirzynski (Ed.), *Software System Design Methodology* (pp. 3-33). Berlin: Springer-Verlag.

Baxter. R, (2004), *'End User Computing Applications, Auditability and other benefits derived from a temporal dimension'*, Proceedings of EUSPRIG 2004 – Risk reduction in End User Computing -'Best practice for spreadsheet users in new Europe', Klagenfurt, Austria, pp 67-70, 1-902724-94-1

Berry, M.J.A., and Linoff, G. (1997), *'Data Mining Techniques'*, NY: John Wiley & Sons.

Bishop. B and McDaid. K, (2007), *'An Empirical study of End User Behaviour in Spreadsheet Error Detection and Correction'*, Proceedings of EuSpRIG 2007 – Enterprise Spreadsheet Management: A Necessary Evil?, Greenwich, London, pp 165-176, ISBN 978-905617-58-6

Blum, A. (1992), *'Neural Networks in C++'*, NY: Wiley, ISBN 0471538477

Boger, Z., and Guterman, H. (1997), *"Knowledge extraction from artificial neural network models"*, IEEE Systems, Man, and Cybernetics Conference, Orlando, FL

Borovska. P, (2006), *'Solving the travelling salesman problem in parallel by Genetic algorithm using micro computer clusters'*, 10th International conference on computer systems and technology, pp 11-17

Brent. R, (1991), *'Fast training algorithms for multi layer neural nets'*, IEEE transactions on neural networks, 2, pp 346-354

Brown and Bostrom, (1989), *'Effective management of End User Computing: A total organisation perspective'*, Journal of Management Information systems, 6, (2), pp 183-212

Brown. P and Gould. J, (1989), *'An experimental study of people creating spreadsheets'*, ACM transactions on information systems, 5 (3), pp253-272

Burgess. C, (1998), *'A tutorial on Support Vector Machines for pattern classification'*, Data mining and knowledge discovery, 2, pp 121-167

Burnett. M, Cook. C, Rothermel. G, (2004) *'End-User Software Engineering'* Communications of the ACM, Sept. 2004, pp 53-58

Burnett. M. Cook. C. Pendse. P. Rothermel.G. Summet. J. Wallace. C, (2003), *'End User Software Engineering with assertions in the spreadsheet paradigm'*, Proceedings of International conference on software engineering, May 2003, Corvallis Oregon, pp33-59

Burnett. M. Rothermel. G. Lixin. L. Dupis. C, Sheretov. A., (2001), *'A methodology for testing spreadsheets'*, ACM transactions on software engineering and methodology, 10 (1), pp 110-147

Bush, M, (1990), "*Formal Inspection Processes—Do They Really Help,"* NSIA Sixth Annual National Joint Conference on Software Quality and Productivity," Williamsburg, VA. Cited in Strauss & Ebenau (1994)

Butler, R. (2000), *'Risk assessment for spreadsheet developments: choosing which models to audit'*, EuSpRIG 2000 Symposium, Spreadsheet Risks—the Hidden Corporate Gamble, pp. 47-56. Greenwich, London

Butler. R and Croll. G, (2006), *'Spreadsheets in clinical medicine – a public health warning'*, Proceedings of EuSpRIG 2006 – Managing spreadsheets: Improving corporate performance, compliance and governance, Greenwich, London, pp 7 – 16, ISBN 1-905617-08-9

Callan. R, (2003), *'Artificial Intelligence'*, 1st edition, Palgrave McMillan, Basingstoke, ISBN 0-333801-36-9

Campbell. D and Stanley. J, (1963), *'Experimental and Quasi experimental designs for research'*, Houghton Mifflin Company, 0-395-30787-2

Cawsey. A, (1998) *'The essence of Artificial Intelligence'*, First edition, Prentice Hall Europe, ISBN 978-0135717790

Chan and Storey (1996), *'The use of spreadsheets in organisations: determinates and consequences'*, Information and Management, 31, pp 119-134

Chan. E and Lippmann. R, (1990), *'Using Genetic Algorithms to Improve Pattern Classification Performance'*. Neural Information Processing systems 1990, pp 797-803

Chedru, F., & Geschwind, N., (1972), *'Writing Disturbances in Acutely Confusional States'*, Neuropsychologia, Volume 10, pp 343-353

Chen. P (1975), *'The entity relationship model – towards a unified view of data'*, Transactions of the ACM, 17 (4), pp 9-36

Clermont, M., Hanin, C. & Mittermeier, R. (2000), *'A spreadsheet auditing tool evaluated in an industrial context'*, EuSpRIG 2000 Symposium, Spreadsheet Risks—the Hidden Corporate Gamble, pp 35-46, Greenwich, London

Clermont. M and Mittermeier. R, (2002), *'Finding High-Level Structures in Spreadsheet Programs'*, 9th Working Conference on Reverse Engineering (WCRE 2002), pp 221-232.

Clermont. M and Mittimeir. R, (2003), *'A Pattern Based Approach to Spreadsheet Auditing'*, Proceedings of the 6th International Conference on Information Systems Implementation and Modelling, Frankfurt, Germany

Clermont. M, (2003), *'A scalable approach to spreadsheet visualisation'*, Unpublished PhD thesis, Available from Universitaet Klagenfurt, Austria

Clermont. M, (2004), *'A toolkit for scalable spreadsheet visualisation'*, Proceedings of EUSPRIG 2004 – Risk reduction in End User Computing -'Best practice for spreadsheet users in new Europe', Klagenfurt, Austria, pp 45-52, 1-902724-94-1

Cochran, W.G., (1977), *'Sampling techniques'*, (3rd edition), New York: Wiley

Colver. D, (2007), *'Inclusion analysis: Finding omission errors'*, Proceedings of EuSpRIG 2007 – Enterprise spreadsheet management: A necessary evil?, Greenwich, London, ISBN 978-905617-58-6

Coopers and Lybrand, (1997), Internet www.planningobjects.com/jungle1.htm, status: not available.

Cragg, P. G. and King, M, (1993*), 'Spreadsheet Modelling Abuse: An Opportunity for OR?'*, Journal of the Operational Research Society, 44(8), pp 743-752.

Craven M.W, Shavlik J.W, (1998), *'Using Neural Networks for Data Mining'*, Future Generation Computer Systems, Volume 13 (2), pp. 211-229

Darlington. K, (2000), *'The essence of Expert Systems'*, Pearson Education, Edinburgh, ISBN 0-13-022774-9

Davies. N. and Ikin. C., (1987), *'Auditing Spreadsheets'* Australian Accountant, December 1987, pp. 54-56.

Davis. G, (1987), *'Commentary on Information systems'*, Accounting horizons, 43, pp103 -105

Dokur. Z, Olmez. T, Yazgan. E, (1997), *'Classification of ECG waveforms by using Genetic Algorithms'*, Proceedings of the 19th Annual International IEEE conference on Engineering in Medicine and Biology, 1 (30), pp 92-94

Dupuis. J.Y. and Wang. F, (2001), *'The cardiac anaesthesia risk evaluation score'*, Anaesthesiology, 94, pp 194-204

EuSpRIG, (2006), *'spreadsheet horror stories'*, Internet http://www.eusprig.org/stories.htm, Online, accessed 23.6.06 12.56pm

Fagan, M. E. (1976), *'Design and Code Inspections to Reduce Errors in Program Development'*, IBM Systems Journal, 15(3), 182-211.

Fagan, M. E., (1986), *'Advances in Software Inspections'*, IEEE Transactions on Software Engineering, 12 (7), pp 744-751

Fisher, R.A., (1922), *'On the interpretation of χ^2 from contingency tables, and the calculation of P'*, Journal of the Royal Statistical Society, 85(1), pp 87-94.

Flood. D and McDaid.K, (2007), *'Voice controlled debugging of spreadsheets'*, Proceedings of EuSpRIG 2007 – Enterprise spreadsheet management: A necessary evil?, Greenwich, London, pp 155 – 165, ISBN 978-905617-58-6

Floyd, B. D. and Pyun, J. (1987), *'Errors in Spreadsheet Use'*, New York: Center for Research on Information Systems, Information Systems Department, New York University.

Forman. G and Cohen. I, (2004), *'Learning from Little: Comparison of Classifiers Given Little Training'*, 3202, pp 161-172

Fraser. J and Smith. P, (1992), *'A catalogue of errors'*, International Journal of man-machine studies, 37, (3), pp 265-307

Galletta, D. F., Hartzel, K. S., Johnson, S., and Joseph, J. L, (1997), *'Spreadsheet Presentation and Error Detection: An Experimental Study'*, Journal of Management Information Systems 13 (2), pp. 45-63

Galletta, D.F.; Abraham, D.; El Louadi, M.; Lekse, W.; Pollailis, Y.A.; & Sampler, J.L, (1993), *"An Empirical Study of Spreadsheet Error-Finding Performance,"* Journal of Accounting, Management, and Information Technology (3:2), pp. 79-95

George, B. and Williams, L. (2003), *'An initial investigation of test driven development in industry'*. Proceedings of the 2003 ACM Symposium on Applied Computing, pp 1135—1139

Gilovitch. T, Griffin. D, Kahneman. D, (2002), *'Heuristics and biases, the psychology of intuitive judgement'*, Cambridge University Press, New York, ISBN 0-521-79679

Graden, M., & Horsley, P. (1986), *'The Effects of Software Inspection on a Major Telecommunications Project'*, AT&T Technical Journal, 65, pp 132 – 147.

Gross. D, (1988), *'Induction and ID/3: More powerful than we think'*, Expert Systems, 5 (4), pp348–348

Grossman, T.A. (2002), *'Spreadsheet Engineering: A research Framework'*, European Spreadsheet Risks Interest Group, 3rd Annual Symposium, Cardiff, UK pp21-34

Grossman. T and Ozluk. O, (2004), *'A Paradigm for Spreadsheet Engineering Methodologies '*, Proceedings of EUSPRIG 2004 – Risk reduction in End User Computing -'Best practice for spreadsheet users in new Europe', Klagenfurt, Austria, pp 23-24, 1-902724-94-1

Grudin, J. (1983), *'Error Patterns in Skilled and Novice Transcription Typing'*, In W. E. Cooper (Ed.), *Cognitive Aspects of Skilled Typewriting* (pp. 121-143). New York: Springer-Verlag

Hasegawa. M and Umeno. K, (2008), *'Solvable Performances of Optimization Neural Networks with Chaotic Noise and Stochastic Noise with Negative Autocorrelation'*, Lecture Notes in Computer Science, 4984, pp 693-704

Haykin. S, (1999), *'Neural Networks a comprehensive foundation'*, Prentice Hall publishers, 2nd Edition, New York, ISBN 0-13-908385-5

Heiber. M (2007), *'Quine-McCluskey Simplification Algorithm'*, Internet http://134.193.15.25/vu/course/cs281/lectures/simplification/quine-McCluskey.html, online accessed 18.0.2007 13.32pm

Hicks and Panko, (1995), *'Capital Budgeting Spreadsheet Code Inspection at NYNEX'*, Internet http://panko.cba.hawaii.edu/ssr/Hicks/HICKS.HTM, 12.1.05, 12.00, Available

Hicks, (1995), cited in Panko. R, (1999), *'What we know about spreadsheet errors'*, Journal of End User Computing, Special issue: Scaling up End User Development

Horvath. A, (2003), Cited in Suykens. J, Brabanter. J, Lukas. L, Vandewalle. J, (2002) *'Weighted Least Squares Support Vector Machines: robustness and sparse approximation'*, Neurocomputing, pp 85-105

Hotopf, N., (1980), *'Slips of the Pen'*, Cognitive Processes in Spelling, pp. 287-307

Howe. H, Simkin. M, (2006), *'Factors affecting the ability to detect spreadsheet errors'*, Decision Sciences Journal of Innovative Education, (4), 1, pp 101-122 http://www.kpmg.co.uk/uk/services/manage/press/970605a.html, Unavailable

Jackson, M. A, (1975), *'Principles of Program Design'*, Academic Press.

Javrin. D and Morrison. J, (1996), *'Factors Influencing Risks and Outcomes in End-User Development'*, Proceedings of the Twenty-Ninth Hawaii International Conference on Systems Sciences, Vol. II, Hawaii, IEEE Computer Society Press, pp. 346-355.

Javrin. D and Morrison. J, (2000), *'Using a structured design approach to reduce risks in End User Spreadsheet development'*, Information & management, 37, pp 1-12

Jenne. S, (1996), *'Audits of End User Computing'*, Internal Auditor, 53 (6), pp 30-35

Jones, T. C., (1998), *'Estimating Software Costs'*, New York, McGraw-Hill.

Jun. S, Pearlmutter. V, Nolte. G, (2002), *'Fast accurate MEG source localization using a multilayer perceptron trained with real brain noise'*, Physics in Medicine and Biology, 47, pp 2547-2560

Karnaugh. M, (1953), *'The Map Method for Synthesis of Combinational Logic Circuits'*, Transactions of American Institute of Electrical Engineers, 72 (9) pp593-599.

Kim. H, and Shin. K, (2007), *'A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets'*, Applied Soft Computing, 7 (2), pp 569-576

Koriat. A, Lichtensein. S, Fischhoff. B, (1980), *'Reasons for overconfidence'*, Journal of experimental psychology: Human learning and memory, 6, pp 107-117

KPMG, (1997), *'Supporting the Decision Maker - A Guide to the Value of Business Modelling'*, Internet

Kruck. S. Maher.J. Barkhi. R , (2003), *'A Framework for Cognitive Skill acquisition and spreadsheet training'*, Journal of End User Computing, 15 (1), pp 20-37.

Leung. M. T., Daouk. H, Chen. A , (2000), *'Forecasting stock indices: a comparison of classification and level estimation models'*, International Journal of Forecasting, 16(2), pp 173-190.

Luger. G, (2005), *'Artificial Intelligence: Structures and Strategies for Complex Problem Solving'*, 5th edition, Addison Wesley, Reading USA, ISBN 978-0321263186

Lukasik. T, (1998), cited in Panko. R, (1999), *'What we know about spreadsheet errors'*, Journal of End User Computing, Special issue: Scaling up End User Development, pp 15-22

Lusted. L, (1977), *'A study of the efficiency of diagnostic radiological procedures: Final report on diagnostics efficiency'*, Chicago: Efficacy Study Committee of the American College of Radiology

Madahar. M, Cleary. P, Ball. D, (2007), *'Categorisation of spreadsheet use within organisations: A progress report'*, Proceedings of EuSpRIG 2007 – Enterprise spreadsheet management: A necessary evil?, Greenwich, London, pp 37-47, ISBN 978-905617-58

Maylor. H and Blackmon. K, (2005), *'Researching Business and Management'*, First edition, Palgrave Macmillan, Basingstoke, UK, ISBN 0-333-96407-1

McDaid. K and Fitzgerald. G, (2008), Private discussions on the use of EDM for overcoming errors of omission world and semantic knowledge in traditional requirements analysis.

McNemar Q., (1947), *'Note on the sampling error of the difference between correlated proportions or percentages'*. Psychometrika, 12, 153-157

MedAL, (2007), *'Cardiac Anesthesia Risk Evaluation (CARE)* spreadsheet, Internet,http://www.medal.org/visitor/www/Active/ch31/ch31.01/Sheets/cardiac%20anesthesia%20risk%20CARE.xls, available, accessed 11.12.07, 4.22pm

Michie. D, (1979), *' Machine models of perceptual and intellectual skills'*, Scientific models and man, Herbert Spencer Lectures, Oxford Science Publications, 0-19857168-2

Michie. D, (1982), *'Mind like capabilities in computers'*, Cognition, 12 (1), pp97-108

Michie. D, (1990), *'Human and machine learning of descriptive concepts'* ICOT Journal. March, pp. 11-20

Michie. D, Muggleton. S, Bain. M, Hayes-Michie.J, (1989), *'An experimental comparison of human and machine learning formalisms'*, Procedings of the 6th International conference of machine learning, pp113-119

Miller.G. A, (1956), *'The magical number seven, plus or minus two'*, Psychological review, 63, pp 81-97

Minsky. M, Papert. S, (1988), *'Perceptrons: An Introduction to Computational Geometry (Expanded Edition)'*, MIT Press, Cambridge, MA, ISBN:0-262-63111-3

Mitchell. M, (1998), *'An Introduction to Genetic Algorithms (Complex Adaptive Systems)'*, MIT Press, Massachusetts, USA, ISBN 978-0262631853

Mitton, R., (1987), *'Spelling Checkers, Spelling Correctors, and the Misspellings of Poor Spellers'*, Information Process Management, 23(5), 495-505

Mookerjee. V, (2001), *'De-biasing Training Data for Inductive Expert System Construction'*, IEEE Transactions on Knowledge and Data Engineering, 13 (3), pp 497 - 512

Muggleton. S, (1991), *'Inductive Logic Programming'*, New Generation Computing, 8(4), pp 295-318

Muggleton. S, (1994), *'Inductive Logic Programming: Derivations, Successes and Shortcomings'*, SIGART Bulletin, 5(1), pp 5-11

Muggleton. S, (1995), *'Inverse Entailment and Progol'*. New Generation Computing. 13 (3&4), pp 245-286

Muggleton. S, (1997), *'Inductive Logic Programming'*, 6th International Workshop, ILP-96, Stockholm, Sweden, pp 145-157

Muggleton. S, Buntine. W, (1988), *'Machine Invention of First Order Predicates by Inverting Resolution'*, Proceedings of the 5th International Workshop on Machine Learning, pp339-352

Muller. M and Padberg. F, (2007), *'About the return on investment on Test Driven Development'*, International Workshop on Economics-Driven Software Engineering Research (EDSER), Portland, Oregon, USA

Munro.M, Huff.S, Moore.G., (1987), *'Expansion and control of End User Computing'*, Journal of management information systems, 4, (3), pp 23-28

Napier. A. Batsell. R. Lane. D. Guadagno. N., (1992), *'Knowledge of command usage in a spreadsheet program'*, Database, 23 (1), pp 13-21

Napier. A. Lane. D. Batsell. R. Guadango. N., (1989), *'Impact of restricted natural language interface on ease of learning and productivity'*, Communications of the ACM, 32 (10), pp 1190-1198

Nash. J and Goldberg. J , (2005), *'Why, how and When spreadsheet tests should be used'*, Proceedings of EUSPRIG 2005 – Managing spreadsheets in the light of Sarbanes-Oxley, London, UK, pp 82-94, ISBN 1-902724-16-X

Nash. J, (2003), *'Audit Change Analysis of Spreadsheets'*, Proceedings of EUSPRIG 2003 – Building better spreadsheets – 'from the ad-hoc to the quality engineered', Dublin, Ireland, 1-86166-199-1

Neurosolutions, (2007), *'Neurosolutions'*, Internet http://www.neurosolutions.com/products/ns/, available, accessed 17.10.07 12.39pm

Newell. A and Simon. H, (1972), *'Human problem solving'*, Englewood Cliffs, New Jersey, Prentice-Hall publishers

Nixon. D and O'Hara. M, (2001), *'Spreadsheet auditing software'*, Proceedings of EUSPRIG 2001 – Spreadsheet Risks, Audit and Development models, London, UK

Norman. D, (1980), *'Categorisation of action slips'*, Psychological review, 88, pp1-15

Nwana. H and Ndumu, (1999), *'A perspective on Software Agent Research'*, The knowledge engineering review, 14, pp 125-142

Oskamp, (1965), *'overconfidence in case-study judgements'*, The journal of consulting psychology, 29, pp 261-265

Oxford English Dictionary (2006), *'Definition of overconfidence'*, Internet http://dictionary.oed.com/cgi/entry/00336672?single=1&query_type=word&queryword=Overconfidence&first=1&max_to_show=10, Online, Accessed 23.6.2006, 10.25am

Oxford English Dictionary (2007), *'Definition of bias'*, Internet http://dictionary.oed.com/cgi/entry/50021538?query_type=word&queryword=Bias&first=1&max_to_show=10&sort_type=alpha&result_place=2&search_id=teXC-a9o3bp-4475&hilite=50021538, Online, Accessed 20.9.2007, 15.23pm

Paine. J, (2001) *'Ensuring Spreadsheet Integrity with Model Master'*, EuSpRIG 2001 Symposium,. Amsterdam, Netherlands, Greenwich University Press.

Paine. J, (2004), *'Spreadsheet Structure Discovery with Logic Programming'*, Proceedings of EUSPRIG 2004 – Risk reduction in End User Computing -'Best practice for spreadsheet users in new Europe', Klagenfurt, Austria, pp 121-134, pp 121-134, ISBN 1-902724-94-1

Paine. J, (2005), *'Excelsior: Bringing the benefits of modularity to Excel'*, Proceedings of EUSPRIG 2005 – Managing spreadsheets in the light of Sarbanes-Oxley, Greenwich, London, pp 161 - 173, ISBN 1-902724-16-X

Paine. J, (2007), *'Practical experience with Excelsior'*, Proceedings of EuSpRIG 2007 – Enterprise spreadsheet management: A necessary evil?, Greenwich, London, pp 131-143, ISBN 978-905617-58-6

Paine. J, Tek. E, Williamson. D, (2006), *'Rapid spreadsheet reshaping with Excelsior: multiple drastic changes to content and layout are easy when you represent enough structure'*, Proceedings of EuSpRIG 2006 – Managing spreadsheets: Improving corporate performance, compliance and governance, Greenwich, London, pp 129 – 147, ISBN 1-905617-08-9

Panko. R and Halverson. R, (1997), *'Spreadsheets on Trial: A Survey of Research on Spreadsheet Risks'*, Hawaii International Conference on System Sciences, Maui, Hawaii, Jan. 2-5, 1996. 326-335

Panko. R and Halverson. R, (1998), *'Are Two Heads Better than One? (At Reducing Errors in Spreadsheet Modelling?'* Office Systems Research Journal, 15 (1), pp. 21-32.

Panko. R, (1999), *'What we know about spreadsheet errors'*, Journal of End User Computing, Special issue: Scaling up End User Development, pp 15-22

Panko. R, (2003), *'Reducing overconfidence in spreadsheet development'*, EUSPRIG Building better spreadsheets from the ad-hoc to the quality engineered', pp 49-57, 1-86166-199-1

Panko. R, (2005),*'Basic Error Rates'*, Internet http://panko.cba.hawaii.edu/HumanErr/Index.htm, Available, accessed on 10.6.05 12.34pm

Panko. R, (2006), *'Recommended practices for spreadsheet testing'*, Proceedings of EuSpRIG 2006 – Managing spreadsheets: Improving corporate performance, compliance and governance, Greenwich, London, pp 73 – 85, ISBN 1-905617-08-9

Panko. R, (2007), *'Thinking is bad: Implications of Human Error Research for Spreadsheet Research and Practice'*, Proceedings of EuSpRIG 2007 – Enterprise Spreadsheet Management: A Necessary Evil?, Greenwich, London, pp 69-81, ISBN 978-905617-58-6

Parsons. H.M., (1974), *"What happened at Hawthorne?"* Science, 183, pp.922-932

Pemberton. J and Robson. A, (2000), *'Spreadsheets in business'*, Industrial management and systems, 100 (8), pp 379-388

Plonsky. M, (2006), *'Analysis of variance – one way'*, Internet http://www.uwsp.edu/psych/stat/12/anova-1w.htm#VI, Available, Accessed 31.7.2008 15.33

Plutowski. M, Cotterell. G, White. H, (1994), *'Learning Mackey Glass from 25 examples, Plus or minus 2'*, Neural Information Processing Systems

Prechelt. L, (1994), *'PROBEN1 – A set of Neural Network Benchmark Problems and Benchmarking tools'*, Technical report 21/94 Fakultat fur informatik, Universitat Karlsrube, Germany

Pressman. R and Ince. D, (2000), *'Software Engineering: A practitioners approach'*, European Adaptation, McGraw Hill

Principe. J.C., Euliano. N. R., Lefebvre. W.C., (2000). *'Neural and adaptive systems: Fundamentals through simulation'*, John Wiley and Sons, New York, ISBN 0-471-35167-9

Pryor (2003), *'Correctness is not enough'*, EUSPRIG Building better spreadsheets from the ad-hoc to the quality engineered', Dublin, Ireland pp 12-24, 1-86166-199-1

Pryor. L, (2004), *'When, Why and how to test spreadsheet models'*, Proceedings of EUSPRIG 2004 – Risk reduction in End User Computing -'Best practice for spreadsheet users in new Europe', Klagenfurt, Austria, pp 23-35, 1-902724-94-1

Purser and Chadwick, (2006), *'Does awareness of differing types of spreadsheet errors aid end-users in identifying spreadsheet errors?'*, Proceedings of EuSpRIG 2006 – Managing spreadsheets: Improving corporate performance, compliance and governance, Greenwich, London, pp 185-204 ISBN 1-905617-08-9

Quinlan. J. R, (1990), *'Learning logical definitions from relations'*, Machine Learning, 5, pp 266-281

Quinlan. J. R., (1986), *'Induction of Decision Trees'*, Machine Learning, (1), pp 81-106

Rajalingham. K, Chadwick. D, Knight. B & Edwards. D, (2000), *'A spreadsheet engineering methodology'*, Proceedings of the Thirty-Third Hawaii International Conference on System Sciences, Maui, Hawaii.

Rajalingham. K., (2005), *'A revised classification of spreadsheet errors'*, Proceedings of EUSPRIG 2005 – Managing spreadsheets in the light of Sarbanes-Oxley, London, UK, pp 185-200, ISBN ISBN 1-902724-16-X

Rasmussen, J., (1974), *"Mental Procedures in Real-Life Tasks: A Case Study of Electronic Troubleshooting."*, Ergonomics, 17(3), pp 293-307

Reason. J, (1990), *'Human Error'*, Cambridge University Press, Cambridge, ISBN 0-521-31419-4

Reason. J, (2005), *'Safety in the operating theatre – Part 2: Human Error and Organisational Failure'*, Quality and Safety in Health Care, 14 (1), pp 56-61

Rummelhart. D and McCelland. J, (1988), *'Parallel Distributed Processing explorations in microstructure processing'*, MIT press, Cambridge Massachusetts, Volume 1, ISBN 0-262-68053-X

Russel. S and Norvig. P, (2003), *'Artificial Intelligence – A Modern Approach'*, 2nd Edition, Pearson education inc., New Jersey, ISBN 0-13-080302-2

Russo. J and Shoemaker. P, (1989), *'Decision traps'*, Simon and Schuster, New York.

Rust. A, Bishop. B, McDaid. K, (2006), *'Investigating the potential of Test-Driven Development for spreadsheet engineering'*, Proceedings of EuSpRIG 2006 – Managing spreadsheets: Improving corporate performance, compliance and governance, Greenwich, London, pp 95-107, ISBN 1-905617-08-9

Rychetsky. M, (2001), *'Algorithms and Architectures for Machine Learning based on Regularized Neural Networks and Support Vector Approaches'*, Shaker Verlag.

Saunders. M, Thornhill. A, Lewis. P, (2007), *'Research designs for business students'*, 4th edition, Pearson Education Limited, Edinburgh, UK, ISBN 9780273701484

Schultheis, R. and Sumner, M, (1994), *'The Relationship of Application Risks to Application Controls: A Study of Microcomputer-Based Spreadsheet Applications'* Journal of End User Computing, 6, pp 11-18.

SERP, (2006), *'Spreadsheet Engineering Research Project'*, Tuck Business School, Dartmouth College, USA

Setiono. R, Kheng Leow. W, Thong. J, (2000), *'Opening the neural network black box: an algorithm for extracting rules from function approximating artificial neural networks'*, Proceedings of the twenty first international conference on Information systems, Brisbane, Australia, pp 176 - 186

Sexton. R and Dorsey. R, (2000), *'Reliable classification using neural networks: a genetic algorithm and backpropagation comparison'* , Decision Support Systems, 30 (1), pp 11-22

Shadish. W, Cook. T, Campbell. D, (2002), *'Experimental and Quasi experimental designs for generalised causal inference'*, 1st Edition, Houghton Mifflin Company, Boston, USA, 0-395-61556-9

Shawe-Taylor. J and Cristianini. N, (2002), *'Support Vector Machines and Kernel Methods, The New Generation of Learning Machines'*, Artificial Intelligence Magazine, Volume 23 (3), pp 31-41

Sietsma. J, Dow. R, (1991), *'Creating artificial neural networks that generalize'*, Neural Networks, **4,** pp67-79.

Stader. J, (1992), *'Applying Neural Networks'*, 1st edition, Artificial Intelligence Applications Institute, ASIN: B0000COHR1

Swain and Guttman, (1983), *'The operational complexity index: A new method for the global assessment of the human factor impact on the safety of advanced reactors concepts'*, Nuclear engineering and design, Volume 236, Issue 10, pp 1113-1121

Sweller. J (1994). *'Cognitive Load Theory, learning difficulty, and instructional design'*, Learning and Instruction 4, pp 295-312

Swingler, K. (1996), *'Applying Neural Networks: A Practical Guide'*, London: Academic Press, ISBN 0-12-679170-8

Takaki, S. T., (2005), *"Self-Efficacy and Overconfidence as Contributing Factors to Spreadsheet Development Errors."*, Working Paper. Information Technology Management Department, College of Business Administration, 2404 Maile Way, Honolulu, HI, 96822

Taylor. M, Moynihan. P, Wood-Harper. T, (1998), *'End User Computing and information systems methodologies'*, Information systems Journal, 8, pp 85-96

Teo.T and Tan. M, (1999), *'Spreadsheet development and 'what if' analysis, quantitative versus qualitative error'*, Accounting management and Information Technology, 9 (3), pp 141-160

Thorne, S., Ball, D. (2009), *'Spreadsheet errors – a mismatch between man and machine', HICSS Hawaii 2009, IN PRESS*

Thorne, S., Ball, D., (2005a), *'Exploring Human Factors in Spreadsheet Development'*, Proceedings of EUSPRIG 2005 – Managing spreadsheets in the light of Sarbanes-Oxley, Greenwich, London, pp 161 - 173, ISBN 1-902724-16-X

Thorne, S., Ball, D., (2005b), *'Human Factors in End User Development'*, Proceedings of UK Academy for Information Systems (UKAIS), Newcastle, UK.

Thorne, S., Ball, D., (2006a), *'An Alternative View of Spreadsheet Error'*, Proceedings of UK Academy for Information Systems (UKAIS), Cheltenham, UK.

Thorne, S., Ball, D., (2006b), *' Considering functional spreadsheet operator usage suggests the value of Example Driven Modelling for Decision Support Systems'*, Proceedings of EuSpRIG 2006 – Managing spreadsheets: Improving corporate performance, compliance and governance, Greenwich, London, pp 95-107, ISBN 1-905617-08-9

Thorne, S., Ball, D., Lawson, Z., (2007), *' Concerning the feasibility of example driven modelling techniques'*, Proceedings of EuSpRIG 2007 – Enterprise spreadsheet management: A necessary evil?, Greenwich, London, pp 131-143, ISBN 978-905617-58-6

Thorne, S., Madahar, M., Cleary, P., Ball, D., Gosling, C., Fernandez, K., (2003), *'Investigating the use of Software Agents to Reduce the Risk of Undetected Errors in Strategic Spreadsheet Application'*, EUSPRIG 4th annual symposium, Dublin, Ireland.

Thorne. S, Ball. D, Lawson. Z., (2004), *'A novel approach to spreadsheet formulae production and overconfidence measurement to reduce risk in spreadsheet modelling'*, Proceedings of EUSPRIG 2004 – Risk reduction in End User Computing, Klagenfurt, pp 71-85, ISBN 1 902724 94 1

Vemula, V.R., Ball, D., Thorne. S, (2006), *' Towards a Spreadsheet Engineering'*, Proceedings of EuSpRIG 2006 – Managing spreadsheets: Improving corporate performance, compliance and governance, Greenwich, London, pp 95-107, ISBN 1-905617-08-9

Walchenbach. J, (2005), *'Microsoft Excel'*, Internet http://j-walk.com/ss/excel/, Available, Accessed on 12.1.05, 11.23am

Wang. D, Dowell. F, Ram. M, Schapaugh. W, (2004), *'Classification of fungal damaged seeds using near infrared spectroscopy'*, International Journal of Food Properties, 7 (1), pp 75-82

Wason, P.C. (1960), *'On the failure to eliminate hypotheses in a conceptual task'*, Quarterly Journal of Experimental Psychology, 12, pp 129-140.

Weiner, E. L, & Nagel, D.C. (1988). *'Human factors in aviation'*, NY: Academic Press

Wing, A. M., & Baddeley, A. D., (1980), *'Spelling Errors in Handwriting: A Corpus and Distributional Analysis',* Cognitive Processes in Spelling (pp. 251-285). London: Academic Press

Yang. Z and Hamer. R, (2007), *'Bio-bass function Neural Networks in protein data mining'*, Current Pharmaceutical Design, 13 (14), pp. 1403-1413

Yirsaw. A, (2003), *'Spreadsheet debugging'*, Proceedings of EUSPRIG 2003 – Building better spreadsheets – 'from the ad-hoc to the quality engineered', Dublin, Ireland, 1-86166-199-1

Yu. L, Wang. S, Lai. K, (2008), *'Credit risk assessment with multistage neural network ensemble approach'*, Expert systems with applications: An international journal, 34 (2), pp 1434-1444

Zhang. G, (2000), *'Neural Networks for classification: A survey'*, IEEE transactions on Systems, Man and Cybernetics, 30 (4), pp 451-462

Zhang. P, (2007), *'A neural network ensemble method with jittered training data for time series forecasting'*, Information Sciences, 177 (23), pp 5329-5346

Zhou. Z and Jiang. Y, (2003), *'Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble'*, IEEE Transactions on Information Technology in Biomedicine, 7 (1), pp 37-42

Appendix A

Chi squared calculations

What follows is the complete list of Chi squared calculations for tasks 1 to 5 summarised in table 3.8.

Task 1 chi squared

The calculations for the chi squared statistic are as follows:

Participants	Success	Failure	Total
Example	20 (a)	5 (b)	25
Control	15 (c)	8 (d)	23
Total	36	12	48

Table A1 Chi squared Task 1

$X^2 = (ad-bc)^2 (a+b+c+d) / (a+b) (c+d) (b+d) (a+c)$

$= 7225*48 /248400$

$= 1.396$

Degrees of freedom = (No. Columns -1) * (No. of Rows -1)

$= (2-1)*(2-1) = 1$

Using the chi squared stat, 1.396, and the degrees of freedom, 1, we look up the value on the chi squared table, see table A2.

Dof	0.5	0.10	0.05	0.02	0.01	0.001
1	0.455	2.706	3.841	5.412	6.635	10.827
2	1.386	4.605	5.991	7.824	9.210	13.815
3	2.366	6.251	7.815	9.837	11.345	16.268
4	3.357	7.779	9.488	11.668	13.277	18.465
5	4.351	9.236	11.070	13.388	15.086	20.517

Table A2 Chi squared critical values

Chi squared (1.396) lies between 0.455 and 2.706

Probability is therefore 0.5 <P< 0.10 i.e. between 50 and 90%

Therefore we accept the null hypothesis.

Task 2 Chi Squared

Participants	Success	Failure	Total
Example	17 (a)	8 (b)	25
Control	13 (c)	10 (d)	23
Total	30	18	48

Table A3 Chi squared Task 2

$X^2 = (ad-bc)^2 (a+b+c+d) / (a+b) (c+d) (b+d) (a+c)$

= 209088 /310500

= 0.673

Degrees of freedom = (No. Columns -1) * (No. of Rows -1)

= (2-1)*(2-1) = 1

The value 0.673 lies in the range 0.455 to 2.706

Probability is therefore 0.5 <P< 0.10 i.e. between 50 and 90%

We accept the null hypothesis

Task 3 chi squared

Participants	Success	Failure	Total
Example	16 (a)	9 (b)	25
Control	10 (c)	13 (d)	23
Total	26	22	48

Table A4 Chi squared task 3

$X^2 = (ad-bc)^2 (a+b+c+d) / (a+b) (c+d) (b+d) (a+c)$

$= 668352/ 328900$

$= 2.032$

Degrees of freedom = (No. Columns -1) * (No. of Rows -1)

$= (2-1)*(2-1) = 1$

The value 2.032 lies in the range 0.455 to 2.706

Probability is therefore $0.5 < P < 0.10$ i.e. between 50 and 90%

We accept the null hypothesis

Task 4 Chi squared

Participants	Success	Failure	Total
Example	16 (a)	9 (b)	25
Control	10 (c)	13 (d)	23
Total	26	22	48

Table A5 Chi squared task 4

$X^2 = (ad-bc)^2 (a+b+c+d) / (a+b) (c+d) (b+d) (a+c)$

$= 668352/ 328900$

$= 2.032$

Degrees of freedom = (No. Columns -1) * (No. of Rows -1)

$= (2-1)*(2-1) = 1$

The value 2.032 lies in the range 0.455 to 2.706

Probability is therefore $0.5 < P < 0.10$ i.e. between 50 and 90%

We accept the null hypothesis

Task 5 Chi squared

Participants	Success	Failure	Total
Example	15 (a)	10 (b)	25
Control	7 (c)	16 (d)	23
Total	22	26	48

Table A6 Chi squared task 5

$X^2 = (ad-bc)^2 (a+b+c+d) / (a+b) (c+d) (b+d) (a+c)$

$= 1387200 / 328900$

$= 4.22$

Degrees of freedom = (No. Columns -1) * (No. of Rows -1)

$= (2-1)*(2-1) = 1$

The value 4.22 lies in the range 3.841 to 5.412

Probability is therefore $0.05 < P < 0.02$ i.e. between 95 and 98%

We reject the null hypothesis

Fisher's exact results

What follows is a complete list of calculations for the Fisher's exact calculations contained in the summary table 3.6.

Fisher's exact for task 1

Fisher's stat for task 1:

	Treatment	Control	Total
Success	20 (a)	15 (b)	
Failure	5 (c)	8 (d)	
			48

Table A7 Fisher's exact task 1

Fisher's exact is in the form: (a+b)! (c+d)! (a+c)! (b+d)! / n! a! b! c! d!

The result of the table is 0.205. This indicates that the probability of this scenario, or a more extreme one occurring is 80%.

We therefore accept the null hypothesis and conclude that the relationship in this example is not significant.

3.5.3.2 Fisher's exact for task 2

	Treatment	Control	Total
Success	17 (a)	13 (b)	
Failure	8 (c)	10 (d)	
			48

Table A8 Fisher's exact task 2

= (a+b)! (c+d)! (a+c)! (b+d)! / n! a! b! c! d!

= 0.301

This indicates that the probability of this scenario, or a more extreme one occurring is 70%.

We therefore accept the null hypothesis and conclude that the relationship in this example is not significant.

3.5.3.3 Fisher's exact for task 3

	Treatment	Control	Total
Success	16 (a)	10 (b)	22
Failure	9 (c)	13 (d)	26
	26	23	48

Table A9 Fishers exact task 3

= (a+b)! (c+d)! (a+c)! (b+d)! / n! a! b! c! d!

= 0. 128

This indicates that the probability of this scenario, or a more extreme one occurring is 88%.

We therefore accept the null hypothesis and conclude that the relationship in this example is not significant.

3.5.3.4 Fisher's exact for task 4

	Treatment	Control	Total
Success	16 (a)	10 (b)	22
Failure	9 (c)	13 (d)	26
	26	23	48

Table A10 Fisher's exact task 4

= (a+b)! (c+d)! (a+c)! (b+d)! / n! a! b! c! d!

= 0. 128

This indicates that the probability of this scenario, or a more extreme one occurring is 88%.

We therefore accept the null hypothesis and conclude that the relationship in this example is not significant.

3.5.3.5 Fisher's exact for task 5

	Treatment	Control	Total
Success	15 (a)	7 (b)	22
Failure	10 (c)	16 (d)	26
	25	23	48

Table A11 Fisher's exact task 5

= (a+b)! (c+d)! (a+c)! (b+d)! / n! a! b! c! d!

= 0.038

This indicates that the probability of this scenario, or a more extreme, i.e. more favourable, one occurring is 96%.

We therefore reject the null hypothesis and conclude that the relationship in this example is significant.

Appendix B

1.0 Full results of reduced training set experiments

Presented in this section are the reduced training set experiments, each size of training set will be dealt with in turn and then all data will be summarised at the end of the section.

1.1 Results of 750 example training set

Presented below are results from the 750 example training set. The 750 example set trained faultlessly.

1.1.1 Confusion matrixes for 750 example T set

Training	Fail	Pass	Merit	Distinction
Fail	**96.1**	**1.9**	**1.9**	**0**
Pass	**0**	**99.1**	**0.97**	**0**
Merit	**0**	**0**	**97.9**	**2.1**
Distinction	**0**	**0**	**0**	**100**

Table B1 Confusion Matrix T values for 750 examples

1.1.2 Confusion matrixes for 750 example CV set

Cross Validation	Fail	Pass	Merit	Distinction
Fail	**96.1**	**1.9**	**1.9**	**0**
Pass	**0**	**99.1**	**0.97**	**0**
Merit	**0**	**0**	**97.9**	**2.1**
Distinction	**0**	**0**	**0**	**100**

Table B2 Confusion matrix CV values for 750 examples

1.1.3 Results for 750 example blind test

Test example number	Desired Fail	Desired Pass	Desired Merit	Desired Distinction	Out Fail	Out Pass	Out Merit	Out Distinction
1	0	1	0	0	-0.03369	0.795127	0.310319	0.200282
2	0	0	1	0	0.237202	0.255117	0.633763	-0.02841
3	0	1	0	0	-0.03373	0.795305	0.309938	0.2007
4	0	0	1	0	0.236296	0.255105	0.631477	-0.02806
5	0	0	1	0	0.234364	0.255453	0.627104	-0.02732
6	1	0	0	0	0.86673	-0.0262	0.170829	0.149967
7	0	0	1	0	0.213478	0.278107	0.632436	-0.02774
8	0	1	0	0	-0.02753	0.755149	0.33773	0.156865
9	0	0	0	1	0.190512	0.136337	0.004971	0.86825

Table B3 Blind test excerpt for 750 example training set

1.2 Results of 500 example training set

Below are the results for the 500 example training set, the 500 example training set trained faultlessly.

1.2.1 Confusion matrix for 500 T set

Training	Fail	Pass	Merit	Distinction
Fail	99.5	0.44	1.9	0
Pass	2.5	94.5	2.5	0
Merit	0.2	11.2	88.6	0
Distinction	0	0	0	100

Table B4 Confusion Matrix T values for 500 examples

1.2.2 Confusion matrix for 500 CV set

Cross Validation	Fail	Pass	Merit	Distinction
Fail	**95.1**	**4.6**	**0**	**0.3**
Pass	**7.8**	**90.3**	**0**	**1.9**
Merit	**1.6**	**3.0**	**85.3**	**10.1**
Distinction	**0**	**0**	**0**	**100**

Table B5 Confusion Matrix CV values for 500 examples

1.2.3 Results for 500 example blind test

Table B6 shows an excerpt of the blind testing results for the 500 example set. Of particular interest is test example number 7, which shows a misclassification of merit as pass. As can be observed the un-normalised probability is marginally in favour of pass.

Test example number	Des Fail	Des Pass	Des Merit	Des Distinction	Out Fail	Out Pass	Out Merit	Out Distinction
1	**0**	**1**	**0**	**0**	**0.043372**	**0.91647**	**0.038368**	**-0.00076**
2	**0**	**0**	**1**	**0**	**-0.01634**	**0.167302**	**0.869585**	**-0.01336**
3	**0**	**1**	**0**	**0**	**0.043372**	**0.91647**	**0.038368**	**-0.00076**
4	**0**	**0**	**1**	**0**	**-0.01798**	**0.102856**	**0.894973**	**0.010139**
5	**0**	**0**	**1**	**0**	**-0.01634**	**0.167302**	**0.869585**	**-0.01336**
6	**1**	**0**	**0**	**0**	**0.908707**	**0.074905**	**-0.0113**	**0.001499**
7	**0**	**0**	**1**	**0**	**-0.01995**	**0.53294**	**0.529982**	**-0.01986**
8	**0**	**1**	**0**	**0**	**-0.02**	**0.534302**	**0.52847**	**-0.01979**
9	**0**	**0**	**0**	**1**	**0.004187**	**-0.00752**	**0.173911**	**0.839199**

Table B6 Blind testing excerpt for the 500 example set

From the full 100 blind tests, only 2 were incorrect. Both tests were merit incorrectly classified as pass. This is consistent with the information contained in both confusion matrixes, tables B6 and B7, which indicate that merit was classified as pass 11.2 and 3% of the time.

1.3 Results of 250 example training set

Below are the results for the 250 example training set, the training process for this set was more challenging.

During training the learning curve was erratic and not smooth, this is an indication of poor training, possibly as a result of having fewer examples to train from. Further, the difference in MSE values for T and CV were more exaggerated, suggesting a poorer ability to generalise.

1.3.1 Confusion Matrix for 250 example T set

Table B7 shows the confusion matrix T results for the 250 example training set. The values indicate that the network has learnt well. In fact table B7 shows that the 250 example training set out performed the larger 500 example training set for the T values, see table B4.

Training	Fail	Pass	Merit	Distinction
Fail	**97.5**	**0.7**	**1.8**	**0**
Pass	**0**	**99.1**	**0.8**	**0**
Merit	**1.8**	**5.6**	**92.4**	**0.2**
Distinction	**0**	**0**	**0**	**100**

Table B7 Confusion matrix T values for 250 training set

1.3.2 Confusion matrix for 250 example CV set

Table B8 shows the confusion matrix CV result for the 250 example training set. In comparison to table B7, the T set, the CV values are significantly lower. This gap between the T and CV values indicates that the network is poorer at generalising to unseen examples. The full extent of this was realised in the blind testing.

Cross Validation	Fail	Pass	Merit	Distinction
Fail	93.2	6.5	0	0
Pass	2.8	96.4	0	0.7
Merit	0	9.4	86.6	3.9
Distinction	0	0	0	100

Table B8 Confusion matrix CV values for 250 training set

1.3.3 Results for 250 example blind test

Table B9 shows an excerpt of blind testing results for the 250 example set, from the full blind testing set there is only 1 misclassification, a pass is misclassified as a fail. Therefore the network correctly classified 99 out of 100 examples.

Test example number	Des Fail	Des Pass	Des Merit	Des Distinction	Out Fail	Out Pass	Out Merit	Out Distinction
1	0	1	0	0	-0.00816	0.96373	0.006479	0.003378
2	0	0	1	0	-0.00406	0.026133	0.951617	0.038397
3	0	1	0	0	-0.00816	0.96373	0.006479	0.003378
4	0	0	1	0	-0.00406	0.026133	0.951617	0.038397
5	0	0	1	0	-0.00406	0.026133	0.951617	0.038397
6	1	0	0	0	0.979259	0.006007	0.005531	-0.00626
7	0	0	1	0	-0.00406	0.026133	0.951617	0.038397
8	0	1	0	0	-0.00816	0.96373	0.006479	0.003378
9	0	0	0	1	-0.04627	-0.00691	0.002792	1.000015

Table B9 Blind testing excerpt for the 250 example set

1.4 Results of 100 example training set

Below are the results from the 100 example training set, observations from the learning process for this set indicates that the network had difficulty learning the problem.

The learning curve was erratic, as the 250 example set was, and the MSE T and CV values were also further apart indicating a worsening ability to generalise.

1.4.1 Confusion matrix result for 100 example T set

Table B10 shows the confusion matrix result for the T set. The values are satisfactory in this matrix, all show a 90% or above accuracy.

Training	Fail	Pass	Merit	Distinction
Fail	**95.6**	**1.4**	**2.9**	**0**
Pass	**0**	**97.2**	**2.8**	**0**
Merit	**2.2**	**6.6**	**90.4**	**0.8**
Distinction	**0**	**0**	**0**	**100**

Table B10 Confusion matrix T results for 100 example training set

1.4.2 Confusion matrix result for 100 example CV set

Table B11 shows the confusion matrix result for the CV set, as with other CV set when compared to their respective T sets, the CV values are lower than the T sets. This indicates a degrading ability to generalise to unseen examples.

This is made clearer when examining the blind testing results.

Cross Validation	Fail	Pass	Merit	Distinction
Fail	**90.3**	**8.6**	**1.1**	**0**
Pass	**3.3**	**95.2**	**0**	**1.5**
Merit	**0**	**10.4**	**84.7**	**4.9**
Distinction	**0**	**0**	**0**	**100**

Table B11 Confusion matrix CV results for 100 example training set

1.4.3 Results for 100 example blind test

Table B12 shows an excerpt of blind testing results for the 100 example set, from the full blind testing set there is only 1 misclassification, a pass is misclassified as a fail. Therefore the network correctly classified 99 out of 100 examples.

Test example number	Des Fail	Des Pass	Des Merit	Des Distinction	Out Fail	Out Pass	Out Merit	Out Distinction
1	0	1	0	0	-0.00816	0.96373	0.006479	0.003378
2	0	0	1	0	-0.00406	0.026133	0.951617	0.038397
3	0	1	0	0	-0.00816	0.96373	0.006479	0.003378
4	0	0	1	0	-0.00406	0.026133	0.951617	0.038397
5	0	0	1	0	-0.00406	0.026133	0.951617	0.038397
6	1	0	0	0	0.979259	0.006007	0.005531	-0.00626
7	0	0	1	0	-0.00406	0.026133	0.951617	0.038397
8	0	1	0	0	-0.00816	0.96373	0.006479	0.003378
9	0	0	0	1	-0.04627	-0.00691	0.002792	1.000015

Table B12 Blind testing excerpt for the 100 example set

1.5 Results for the 50 example training set

Below are the results from the 50 example training set, observations from the learning process for this set indicates that the network had difficulty learning the problem.

The learning curve was erratic, as was the 100 and 250 example set, the MSE T and CV values were also further apart indicating a worsening ability to generalise.

1.5.1 Confusion matrix result for 50 example T set

Table B13 shows the confusion matrix T results, all values are acceptable, all are above 90%.

Training	Fail	Pass	Merit	Distinction
Fail	**92.6**	**2.8**	**4.5**	**0**
Pass	**0**	**96.2**	**2.9**	**0.9**
Merit	**1.1**	**5.9**	**92.5**	**0.5**
Distinction	**0**	**0**	**0.2**	**99.8**

Table B13 Confusion matrix T result for 50 example training set

1.5.2 Confusion matrix result for 50 example CV set

Table B14 shows the confusion matrix CV results, the classifications fail and particularly merit show a drop in accuracy when compared to the T values. Again, this is an indication of a degrading ability to generalise. This becomes more apparent after blind testing is performed.

Cross Validation	Fail	Pass	Merit	Distinction
Fail	**89.6**	**9.5**	**0.9**	**0**
Pass	**3.5**	**94.7**	**0**	**1.8**
Merit	**0**	**9.9**	**85**	**5.1**
Distinction	**0**	**0**	**3**	**97**

Table B14 Confusion matrix CV result for 50 example training set

1.5.3 Results of 50 example blind test

Table B15 shows an excerpt of blind testing results for the 50 example set, from the full blind testing set there are 2 misclassifications. Therefore the network correctly classified 98 out of 100 examples.

Test example number	Des Fail	Des Pass	Des Merit	Des Distinction	Out Fail	Out Pass	Out Merit	Out Distinction
1	0	1	0	0	0.041538	0.910121	0.037494	-0.03168
2	0	0	1	0	0.016333	0.051945	0.811304	0.21035
3	0	1	0	0	0.03912	0.910197	0.038594	-0.03166
4	0	0	1	0	0.016369	0.051714	0.814141	0.211778
5	0	0	1	0	0.016332	0.051949	0.811236	0.210333
6	1	0	0	0	0.803542	0.091342	0.035088	0.031645
7	0	0	1	0	0.016298	0.052229	0.808444	0.208252
8	0	1	0	0	0.035121	0.904779	0.04192	-0.03127
9	0	0	0	1	0.009007	0.035498	0.087635	0.903804

Table B15 Blind testing results for 50 example training set

1.6 Results for the 25 example training set

Below are the results from the 25 example training set, observations from the learning process for this set indicates that the network had difficulty learning the problem.

The learning curve was erratic, as was the 50,100 and 250 example set, the MSE T and CV values were also further apart indicating a worsening ability to generalise.

1.6.1 Confusion matrix for 25 example T set

From table B16 we can see that most of the accuracy figures are acceptable, the only notable exception is merit.

According to table 18 merit is classified as merit 88.8% of the time. The remaining time it is incorrectly classified as Fail (1.7%), Pass (6.1%) and Distinction (3.4%). The true magnitude of this was revealed in the blind testing.

Cross Validation	Fail	Pass	Merit	Distinction
Fail	**91.5**	**3.5**	**5**	**0**
Pass	**0.2**	**95.1**	**3.4**	**1.3**
Merit	**1.7**	**6.1**	**88.8**	**3.4**
Distinction	**0**	**0**	**1.8**	**98.2**

Table B16 Confusion matrix T results for 25 example training set

1.6.2 Confusion matrix result for 25 example CV set

Table B17 shows the CV results for the 50 example training set. As observed in other CV sets, the CV values are lower than the T values, which is normal. However, the classification Fail shows more difference between T and CV values than any other classification. This indicates that the classification fail may have difficulty dealing with unseen examples, this was be revealed in the blind testing.

Cross Validation	Fail	Pass	Merit	Distinction
Fail	**80.5**	**13.4**	**1.7**	**4.3**
Pass	**5.5**	**91.4**	**0**	**3.1**
Merit	**0**	**9.8**	**84.5**	**5.6**
Distinction	**0**	**0.1**	**3.2**	**96.7**

Table B17 Confusion matrix CV results for 25 example training set

1.6.3 Results of 25 example blind test

Table B18 shows an excerpt of blind testing results for the 25 example set, from the full blind testing set there are 4 misclassifications. Therefore the network correctly classified 96 out of 100 examples.

Test example number	Des Fail	Des Pass	Des Merit	Des Distinction	Out Fail	Out Pass	Out Merit	Out Distinction
1	0	1	0	0	0.007175	0.939516	0.027067	0.032472
2	0	0	1	0	0.027413	0.021542	0.932081	0.031685
3	0	1	0	0	0.007193	0.939222	0.027363	0.03249
4	0	0	1	0	0.022805	0.025052	0.924395	0.03451
5	0	0	1	0	0.027364	0.021492	0.932015	0.031793
6	1	0	0	0	0.960472	0.063549	0.025046	-0.03775
7	0	0	1	0	0.027843	0.03414	0.918125	0.028065
8	0	1	0	0	0.043117	0.887032	0.047802	0.022986
9	0	0	0	1	-0.01483	0.016991	0.062292	0.936098

Table B18 Blind testing excerpt for 25 example training set

1.7 Results for the 20 example training set

Below are the results from the 20 example training set, observations from the learning process for this set indicates that the network had difficulty learning the problem.

The learning curve was erratic, as was the 25, 50, 100 and 250 example set, the MSE T and CV values were also further apart indicating a worsening ability to generalise

1.7.1 Confusion matrix result for 20 example T set

The confusion matrix T results, see table B19, show a surprisingly high level of accuracy given Plutoski's assertion that 25 plus or minus 2 is the minimum number required in a training set (Plutoski *et al.,* 1994)

Training	Fail	Pass	Merit	Distinction
Fail	95.3	1.5	3.2	0
Pass	0.1	96.8	2.4	0.7
Merit	1.8	6	89.1	3.1
Distinction	0	0	0.9	99.1

Table B19 Confusion Matrix T results for the 20 example training set

1.7.2 Confusion matrix result for 20 example CV set

The confusion matrix CV results suggest a lower accuracy than table B20, which is not unexpected. Again, this suggests a poorer ability to generalise which was fully realised after the blind testing.

Cross Validation	Fail	Pass	Merit	Distinction
Fail	**81.2**	**12**	**2.4**	**4.4**
Pass	**5.1**	**91.7**	**0**	**3.2**
Merit	**0**	**10.1**	**85.2**	**4.7**
Distinction	**0**	**0.1**	**1.6**	**98.3**

Table B20 Confusion Matrix CV results for the 20 example training set

1.7.3 Results of 20 example blind test

Table B21 shows an excerpt of blind testing results for the 20 example set. In the excerpt there are 2 misclassifications: example number 3 and 5.

From the full 100 blind test examples, the network misclassified 5 and correctly classified 95.

Again considering Plutoski's evidence (Plutoski *et al.,* 1994), this level of accuracy is surprising.

Test example number	Des Fail	Des Pass	Des Merit	Des Distinction	Out Fail	Out Pass	Out Merit	Out Distinction
1	0	1	0	0	0.023359	0.984063	-0.04131	0.011507
2	0	0	1	0	0.056378	-0.02702	0.976597	-0.00293
3	0	1	0	0	0.984041	0.023417	-0.04131	0.01149
4	0	0	1	0	0.01267	0.001327	0.977408	-0.01897
5	0	0	1	0	0.056706	-0.04723	0.550115	0.654901
6	1	0	0	0	0.949295	0.022741	0.038409	0.041742
7	0	0	1	0	-0.04782	0.257387	0.6759	0.067918
8	0	1	0	0	0.020304	0.979883	-0.04003	0.012226
9	0	0	0	1	-0.04723	0.013377	0.03137	0.977877

Table B21 Blind testing excerpt for 20 example training set

1.8 Results for 15 example training set

Below are the results from the 15 example training set, observations from the learning process for this set indicates that the network had severe difficulty learning the problem.

The learning curve was severely erratic, the MSE T and CV values were significantly further apart indicating a worsening ability to generalise.

1.8.1 Confusion matrix result for 15 example T set

Table 26 indicates that the accuracy has significantly dropped for the 15 example training set. Table B22 indicates that for Pass and Distinction the network was unable to correctly classify any examples.

Training	Fail	Pass	Merit	Distinction
Fail	**100**	**0**	**0**	**0**
Pass	**10.2**	**0**	**86.1**	**3.7**
Merit	**0**	**0**	**100**	**0**
Distinction	**0**	**0**	**100**	**0**

Table B22 Confusion matrix T result for 15 example training set

1.8.2 Confusion matrix result for 15 example CV set

Table B23 shows similar results to that of 24, the values contained in the table suggest that the network was unable to classify Pass or Distinction correctly.

Cross Validation	Fail	Pass	Merit	Distinction
Fail	**100**	**0**	**0**	**0**
Pass	**14.5**	**0**	**82.5**	**3**
Merit	**0**	**0**	**100**	**0**
Distinction	**0**	**0**	**100**	**0**

Table B23 Confusion matrix CV results for 20 example training set

Again, further insight was provided by examining the training set.

1.8.3 Result of 15 example blind test

Table B24 shows an excerpt of blind testing results for the 15 example set. In the excerpt alone, table 26, there are 3 misclassifications: example numbers 2, 5 and 8.

From the full 100 blind test examples, the network misclassified 15 and correctly classified 85.

Test example number	Des Fail	Des Pass	Des Merit	Des Distinction	Out Fail	Out Pass	Out Merit	Out Distinction
1	0	1	0	0	0.021125	0.956723	0.007533	0.019255
2	0	0	1	0	-0.00332	0.972343	0.033403	0.013243
3	0	1	0	0	0.022656	0.955189	0.003859	0.021109
4	0	0	1	0	-0.00318	-0.00782	0.973565	0.01112
5	0	0	1	0	0.016637	0.032677	0.067919	0.943713
6	1	0	0	0	0.978956	0.039092	0.019732	0.016902
7	0	0	1	0	-0.00079	0.013454	0.949689	-0.0089
8	0	1	0	0	0.952717	0.019518	0.018239	0.013152
9	0	0	0	1	0.017005	0.052647	0.041858	0.962332

Table B24 Blind testing excerpt for the 15 example training set

Considering the values contained in the confusion matrixes, tables B22 and B23, the network has performed rather well in testing. Further, both confusion matrixes state that the network was unable to correctly classify Pass or Distinction, yet there are several examples of the network correctly classifying these in the blind testing.

1.10 Confusion matrix results of experimentation with reduced training sets.

The following summary statistics are drawn from the confusion matrixes of each different sized training set.

The graphs plot accuracy, T and CV values, for each classification (Fail, Pass, Merit and Distinction) as the number of examples in the training set are reduced.

1.10.1 Accuracy of classification Fail

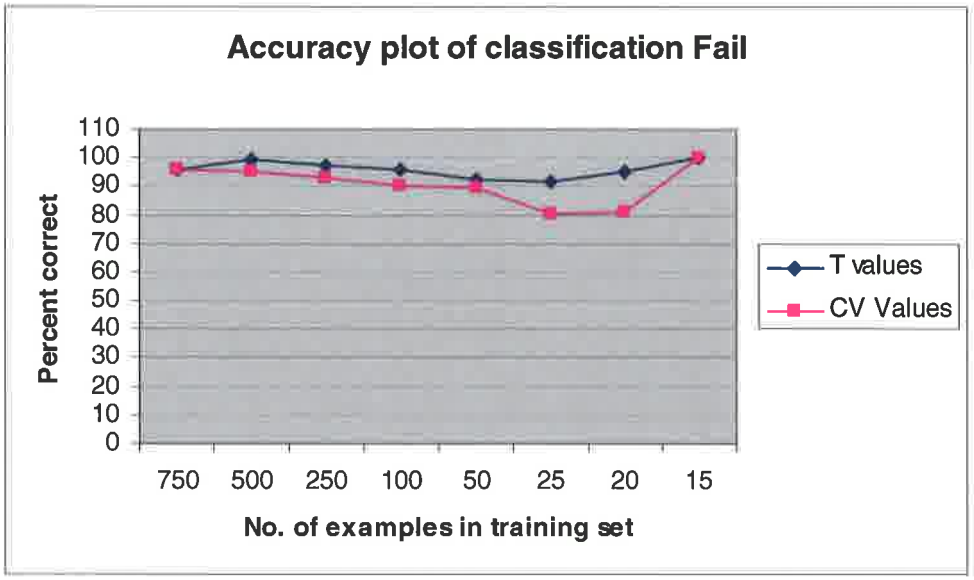

Figure B1 Accuracy plot of classification Fail

Figure B1 shows the accuracy plot for the classification "Fail", as the numbers of examples of the X axis are reduced. Accuracy for the T set doesn't drop below 90% at all, and accuracy for the CV drops below 90% on the 25 and 20 example training sets.

When using the 15 example training set, the confusion matrix results are unusual and suggest that the network has not learnt the problem well. In figure B1 both the CV and T values for the 15 example data set read as 100% accurate. This is a false impression as highlighted by the blind test score for the 15 example training set in figure 5.

1.10.2 Accuracy of classification Pass

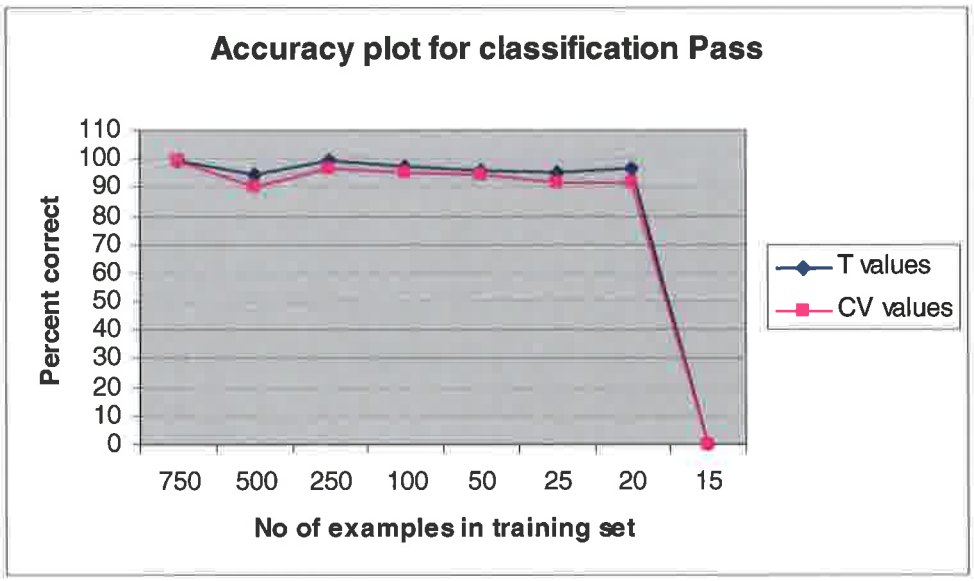

Figure B2 Accuracy plot of classification Pass

Figure B2 shows the accuracy plot for the classification Pass. Accuracy for both the T and CV sets doesn't drop below 90% until reaching the 15 example training set. Also for training sets of sizes 750 through to 20, the CV and T values are closer together for Pass than Fail, see figure B1. This is a desirable result as it suggests that the network is able to generalise "pass" to unseen data with a similar accuracy as achieved with the training set.

As with figure B1, the 15 example training set showed unusual results. The accuracy drops from 96% and 91% (20 example training set, T and CV values respectively) to both T and CV values being 0% when using 15 examples. However, this 0% accuracy is contradicted by the results of the blind testing which shows Pass classifications being correctly classified by the network.

1.10.3 Accuracy of classification Merit

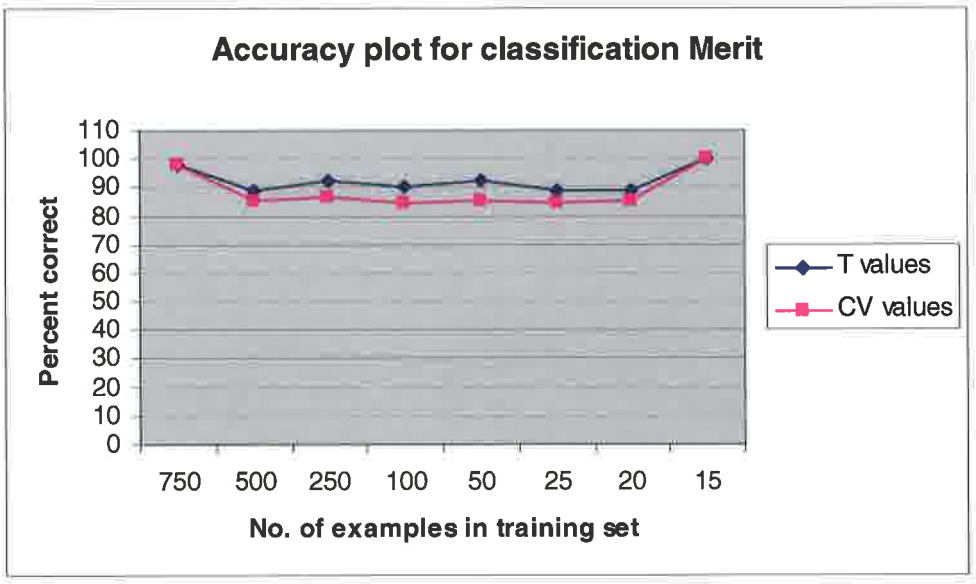

Figure B3 Accuracy plot of classification Merit

Figure B3 shows the accuracy plot for the classification Merit. In comparison to all other classifications, merit shows the poorest performance in two ways.

Firstly the accuracy values for T drop below 90% for the 500, 100, 25 and 20 example training sets. Accuracy values for CV drop below 90% on all sets other than 750 and 15 example training sets.

In addition the gap between the T and CV values are greater, indicating the network has more difficulty generalising "merit" to unseen data.

As with classifications Fail and Pass, the 15 example training set produces unusual results with the classification Merit. As can be seen in the graph the 15 example training set produces an accuracy of 100%, however in the blind testing, figure 5, there are examples of the classification Merit being misclassified, this conflicts with the reported 100% accuracy rate in figure B3.

1.10.4 Accuracy of classification Distinction

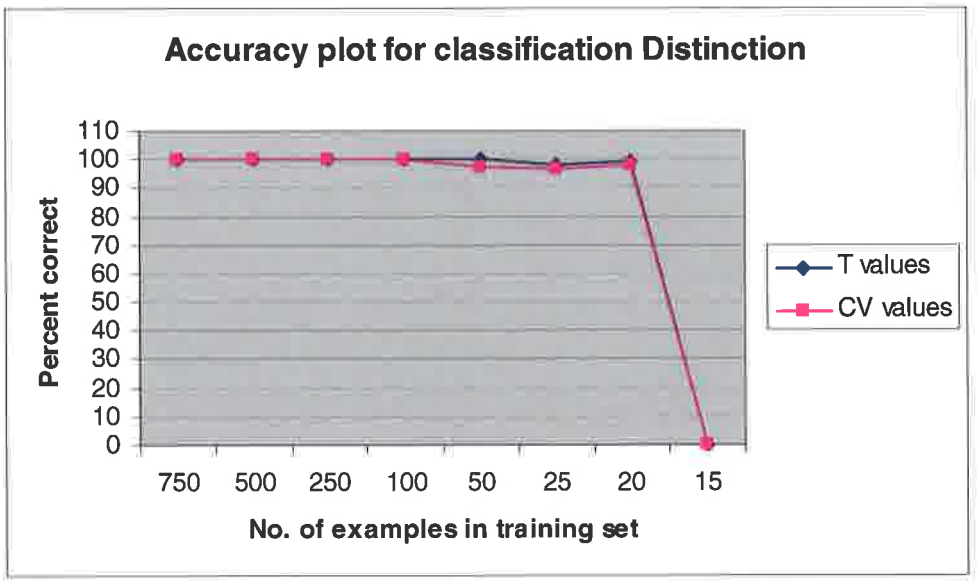

Figure B4 Accuracy plot for classification Distinction

Figure B4 shows the accuracy plot for the classification Distinction. In comparison to all other classifications, Distinction shows the best performance in two ways.

Firstly values for both T and CV do not drop below 95% with the exception of the 15 example training set. In fact the 750, 500, 250 and 100 example training sets show 100% accuracy.

Further, the CV and T values are closer together than for any other classification. This suggests that the network is able to generalise Distinction to unseen data with a similar level of accuracy attained with the training set alone.

2.0 Full results of the complexity experiment

Contained in this section are the full results for the complexity experiments in chapter 5. The results are presented sequentially, detailing full T and CV confusion matrixes and excerpts of blind testing results.

2.1 Results of 2 class experiment

Presented below are results from the 2 class training set. The 2 class training set trained faultlessly.

2.1.1 Confusion matrix 2 class T set

T	Pass	Fail
Pass	100	0
Fail	0	100

Table B25 T value confusion matrix result for 2 class set

As can bee seen above in B25, the confusion matrix for the T set shows a perfect result.

2.1.2 Confusion matrix 2 class CV set

CV	Pass	Fail
Pass	100	0
Fail	0	100

Table B26 CV value confusion matrix result for 2 class set

As can bee seen above in B26, the confusion matrix for the T set shows a perfect result.

2.1.3 Result of 2 class blind testing

Des PASS	Des FAIL	Out PASS	Out FAIL
1	0	1.000003	0.000014
0	1	0.000013	1.000025
1	0	1.000003	0.000014
1	0	1.000003	0.000014
1	0	1.000003	0.000014
0	1	0.000013	1.000025
1	0	1.000003	0.000014
0	1	0.000013	1.000025
1	0	1.000002	0.000014

Table B27 Excerpt of 2 class blind testing result

The classification accuracy for the 2 class blind testing was 100%, i.e. none of the blind test examples were classified in error. See table B27 for an excerpt of this training data.

2.2 Results of 3 class experiment

Presented below are results from the 3 class training set. The 3 class training set trained faultlessly.

2.2.1 Confusion matrix 3 class T set

T	Pass	Fail	Merit
Pass	99.4	0.42	0.2
Fail	0	100	0
Merit	0	0	100

Table B28 T value confusion matrix result for 3 class set

As can be seen in table B28, the T value confusion matrix shows a very positive learning result.

2.2.2 Confusion matrix 3 class CV set

CV	Pass	Fail	Merit
Pass	99.4	0.44	0.19
Fail	0	100	0
Merit	0	0	100

Table B29 CV value confusion matrix result for 3 class set

As can be seen in table B29, the CV value confusion matrix shows a very positive learning result, this is similar to table B28.

2.2.3 Result of 3 class blind testing set

Des PASS	Des FAIL	Des MERIT	Out PASS	Out FAIL	Out MERIT
0	0	1	-0.02457	0.006095	0.996516
0	1	0	0.012072	0.987592	0.005733
0	1	0	0.012614	0.988291	0.004742
0	1	0	0.012615	0.988289	0.004742
0	1	0	0.012614	0.988291	0.004743
1	0	0	0.95795	0.042475	-0.00154
1	0	0	0.955966	0.045336	-0.00267
0	1	0	0.012614	0.988291	0.004743
0	1	0	0.01261	0.988286	0.004749
1	0	0	0.95795	0.042476	-0.00154

Table B30 Excerpt of 3 class blind testing result

The classification accuracy for the 3 class blind testing was 100%, i.e. none of the blind test examples were classified in error. See table B30 for an except of this testing data

2.3 Results of 4 class experiment

Presented below are results from the 4 class training set. The 4 class training set trained well showing only minor error.

2.3.1 Confusion matrix 4 class T set

T	Pass	Fail	Merit	Distinction
Pass	92.1	5.2	2.5	0.2
Fail	0	100	0	0
Merit	0	0	93.8	6.2
Distinction	0	0	0	100

Table B31 T value confusion matrix result for 4 class set

As can be seen in table B31, the T value confusion matrix shows a positive learning result. The only exception to this is the classification Pass which shows some confusion with both Fail and merit

2.3.2 Confusion matrix 4 class CV set

CV	Pass	Fail	Merit	Distinction
Pass	89.4	7.7	2.9	0
Fail	0	100	0	0
Merit	0	0	90.9	9.1
Distinction	0	0	0	100

Table B32 CV value confusion matrix result for 2 class set

As can be seen in table B32, the CV value confusion matrix shows a positive learning result. There is some error on classifications Pass and Merit.

2.3.3 Result of 4 class blind testing set

Out PASS	Out FAIL	Out MERIT	Out DISTIN	Out PASS	Out FAIL	Out MERIT	Out DISTIN
1	0	0	0	0.92275	0.021293	0.045332	-0.02114
0	0	1	0	-0.02367	-0.00998	0.987259	-0.00848
0	0	1	0	-0.02381	-0.00975	0.987494	-0.00867
0	1	0	0	0.042007	0.987415	-0.00967	-0.04461
1	0	0	0	0.930152	0.013113	0.061006	-0.02276
1	0	0	0	0.923025	0.020999	0.045835	-0.0212
1	0	0	0	0.93965	0.000108	0.095343	-0.02497
0	0	1	0	-0.0238	-0.00977	0.987475	-0.00866
0	0	0	1	0.03399	0.035582	0.019871	0.963659
0	0	0	1	0.03399	0.035582	0.019871	0.963659

Table B33 Excerpt of 4 class blind testing result

The classification accuracy for the 4 class blind testing set was 99%, i.e. all but one of the blind tests passed successfully, table B33 contains an excerpt of this data.

2.4 Results of 5 class experiment

Presented below are results from the 5 class training set. The 5 class training set trained well showing only minor error.

2.4.1 Confusion matrix 5 class T set

T	Pass	Fail	Merit	Distinction	Commendation
Pass	96.4	0	3.57	0	0
Fail	3.84	96.1	0	0	0
Merit	0	0	100	0	0
Distinction	0	0	0	93.5	6.5
Commendation	0	0	0	0	100

Table B34 T value confusion matrix result for 5 class set

As can be seen in table B34, the T value confusion matrix shows a positive learning result. There is error present in classifications: Distinction; Fail and Pass.

2.4.2 Confusion matrix 5 class CV set

CV	Pass	Fail	Merit	Distinction	Commendation
Pass	85.6	7.14	7.14	0.2	0
Fail	11.6	88.4	0	0	0
Merit	0	3.9	86.7	6.4	3
Distinction	0	0	14.4	85.5	0.1
Commendation	0	0	0	0	100

Table B35 CV value confusion matrix result for 5 class set

As can be seen in table B35, the CV value confusion matrix shows a moderately positive learning result. The differences between values in B34 and B35 suggest a lessening ability to generalise correctly.

2.4.3 Result of 5 class blind testing set

Des PASS	Des FAIL	Des MERIT	Des DISTIN	Des COMM	Out PASS	Out FAIL	Out MERIT	Out DISTIN	Out COMM
0	0	1	0	0	-0.000623	0.001703	0.974973	-0.01404	0.000857
0	0	1	0	0	-0.00059	0.001644	0.97494	-0.01398	0.000985
0	0	0	1	0	0.07278	-0.040106	0.091911	0.894266	0.0365
1	0	0	0	0	0.965957	0.057118	0.005823	-0.00784	-0.04126
1	0	0	0	0	0.990883	0.033191	0.000837	0.01871	-0.04094
0	0	1	0	0	-0.000575	0.001621	0.974927	-0.01395	0.001033
0	0	1	0	0	0.000155	0.00209	0.975165	-0.01405	-0.00259
0	1	0	0	0	0.004337	1.00221	-0.04137	-0.00699	0.007167
1	0	0	0	0	0.994959	0.029564	-0.0004	0.024721	-0.04087
0	0	0	0	1	-0.054463	0.048293	0.064703	0.079828	0.938982

Table B36 Excerpt of 5 class blind testing result

The classification accuracy for the 5 class blind testing set was 98%, i.e. all but two of the blind tests passed successfully, table B36 contains an excerpt of this data.

2.5 Results of the 6 class experiment

Presented below are results from the 6 class training set. The 6 class training set trained well showing only minor error.

2.5.1 Confusion matrix 6 class T set

T	Pass	Fail	Merit	Distinction	Commendation	Excellence
Pass	100	0	0	0	0	0
Fail	0	100	0	0	0	0
Merit	0		100	0	0	0
Distinction	0	0	0	100	0	0
Commendation	0	0	0	10	90	0
Excellence	0	0	0	0	20	80

Table B37 T value confusion matrix result for 6 class set

As can be seen in table B37, the T value confusion matrix shows a moderately positive learning result. Some confusion is evident between classifications Commendation and Excellence.

2.5.2 Confusion matrix 6 class CV set

CV	Pass	Fail	Merit	Distinction	Commendation	Excellence
Pass	88.2	8.8	2.9	0	0	0
Fail	2.3	97.3	0.3	0	0	0
Merit	0.9	6.3	92.8	0	0	0
Distinction	0	0	0	70.3	20.3	9.2
Commendation	0	0	0	15.7	80.6	3.6
Excellence	0	0	0	4.6	24.7	70.5

Table B38 CV value confusion matrix result for 6 class set

As can be seen in table B38, the CV value confusion matrix shows a moderately positive learning result. The differences between values in B37 and B38 suggest a worsening ability to generalise correctly.

2.5.3 Results of 6 class blind testing set

Des PASS	Des FAIL	Des MERIT	Des DISTIN	Des COMM	Des EXCEL	Out PASS	Out FAIL	Out MERIT	Out DISTIN	Out COMM	Out EXCEL
0	1	0	0	0	0	0.002829	0.995154	0.007147	-0.05052	0.0038	0.012028
0	1	0	0	0	0	0.000566	0.996703	0.006334	-0.05055	0.004803	0.013961
0	0	0	0	0	1	-0.028681	0.045872	0.051064	0.047655	0.493291	0.469094
0	0	1	0	0	0	0.008742	-0.03628	0.957192	0.003313	-0.01921	-0.00685
0	0	1	0	0	0	-0.00243	-0.0379	0.972247	0.011313	-0.01699	-0.00044
0	0	0	1	0	0	-0.002924	-0.03777	0.017051	0.967732	-0.01656	0.000396
0	1	0	0	0	0	0.002708	0.995207	0.007183	-0.05052	0.003782	0.012067
0	1	0	0	0	0	0.000062	0.996879	0.006513	-0.05055	0.004645	0.014049
0	1	0	0	0	0	0.000717	0.99664	0.006308	-0.05055	0.004824	0.013912
0	0	1	0	0	0	-0.001967	-0.03787	0.971711	0.010937	-0.01698	-0.00062

Table B39 Excerpt of 6 class blind testing result

The classification accuracy for the 6 class blind testing set was 98%, i.e. all but two of the blind tests passed successfully, table B36 contains an excerpt of this data.

2.6 Results of 7 class experiment

Presented below are results from the 7 class training set. The 7 class training set trained moderately well but showed signs of problematic learning.

2.6.1 Confusion matrix 7 class T set

T	Compensate	Pass	Fail	Merit	Distinction	Commendation	Excellence
Compensate	85.7	14.3	0	0	0	0	0
Pass	0	97.2	0	2.7	0	0	0
Fail	0	0	100	0	0	0	0
Merit	0	10.5	0	89.9	0	0	0
Distinction	0	0	0.1	3.4	95.7	0.7	0
Commendation	0	0	0	0	0	99.8	0.2
Excellence	0	0	0	0	0.2	2.3	97.4

Table B40 T value confusion matrix result for 7 class set

As can be seen in table B40, the T value confusion matrix shows a moderately positive learning result. However of particular concern are compensate and merit

2.6.2 Confusion matrix 7 class CV set

CV	Compensate	Pass	Fail	Merit	Distinction	Commendation	Excellence
Compensate	71.4	28.5	0	0	0	0	0
Pass	37.6	62.4	0	0	0	0	0
Fail	0	1.5	95.4	3.1	0	0	0
Merit	0	0	12	84.2	3.7	0	0
Distinction	0	0	0	0.5	79.5	10.4	9.5
Commendation	0	0	0	0	3.3	89.3	7.3
Excellence	0	0	0	0	0.8	8.6	90.5

Table B41 CV value confusion matrix result for 7 class set

As can be seen in table B41, the CV value confusion matrix shows a moderately positive learning result. The differences between the B40 and B41 suggest only a moderate ability to generalise.

2.6.3 Results of 7 class blind testing set

Desired	Out PASS	Out FAIL	Out MERIT	Out DISTIN	Out COMM	Out EXCEL	Out COMP
FAIL	-0.053876	0.939954	-0.04621	-0.04532	-0.02259	0.052071	0.045628
PASS	0.899678	-0.03396	0.107112	-0.04069	-0.03726	-0.03212	0.036653
PASS	0.895666	-0.03552	0.119703	-0.04043	-0.03677	-0.0308	0.028163
PASS	0.896468	-0.03527	0.117573	-0.04048	-0.03686	-0.03103	0.029562
PASS	0.762542	-0.04528	0.336641	-0.03502	-0.03409	-0.0131	-0.02871
FAIL	-0.053876	0.939954	-0.04621	-0.04532	-0.02259	0.052071	0.045628
DISTIN	-0.044092	-0.0109	0.106232	0.370242	0.216415	0.149384	-0.05492
MERIT	0.024237	-0.05056	0.819631	0.065025	-0.01385	0.057901	-0.05377
COMP	0.316055	-0.02775	-0.02892	-0.05144	-0.0461	-0.01582	0.455271
EXCEL	-0.054103	-0.01524	-0.03706	0.174913	0.197065	0.40914	-0.05414

Table B42 Excerpt of 7 class blind testing result

The classification accuracy for the 7 class blind testing set was 95%, this is considered the minimum level of accuracy that is acceptable. Table B42 contains an excerpt of this data.

2.7 Results of 8 class experiment

Presented below are results from the 8 class training set. The 8 class training set trained moderately but exhibited erratic learning curves. This suggests some significant learning difficulty.

2.7.1 Confusion matrix 8 class T set

T	Resit	Compensate	Pass	Fail	Merit	Distinction	Commendation	Excellence
Resit	60.3	0	0	39.6	0	0	0	0
Compensate	22.2	55.5	22.2	0	0	0	0	0
Pass	0	0	96	0	4	0	0	0
Fail	0	0	0	100	0	0	0	0
Merit	0	0	4.1	0	91.6	4.1	0	0
Distinction	0	0	0	0	0	100	0	0
Commendation	0	0	0	0	0	20	80	0
Excellence	0	0	0	0	0	0	0	100

Table B43 T value confusion matrix result for 8 class set

As can be seen in table B43, the T value confusion matrix shows a mixed learning result. Of particular concern are compensate, resit and commendation.

2.7.2 Confusion matrix 8 class CV set

CV	Resit	Compensate	Pass	Fail	Merit	Distinction	Commendation	Excellence
Resit	40	60	0	0	0	0	0	0
Compensate	0	33.3	44.4	22.2	0	0	0	0
Pass	0	0	92	0	8	0	0	0
Fail	0	0	0	80.9	19.04	0	0	0
Merit	0	0	0	0	95.8	4.1	0	0
Distinction	0	0	0	0	11.1	88.8	0	0
Commendation	0	0	0	0	0	40	60	0
Excellence	0	0	0	0	0	100	0	0

Table B44 CV value confusion matrix result for 2 class set

As can be seen in table B44, the CV value confusion matrix shows poor learning result. The differences between the B43 and B44 suggest a poor ability to generalise.

2.7.3 Results of 8 class blind testing set

Desired	Out PASS	Out FAIL	Out MERIT	Out DISTIN	Out COMM	Out EXCEL	Out COMP	Out RESIT
FAIL	-0.004909	0.998267	-0.00971	0.019293	0.017171	-0.004684	-0.000206	0.010412
COMP	0.016821	-0.043314	0.007966	-0.052641	-0.037405	-0.048695	0.963236	0.040603
PASS	0.980484	-0.027938	0.030482	0.017888	0.0155	0.014608	-0.010113	0.011675
FAIL	-0.012265	0.991084	-0.011807	0.007725	0.012713	-0.007825	-0.008589	0.011288
COMM	-0.021361	0.065854	-0.039488	0.385957	0.603696	-0.023671	-0.036764	-0.052996
MERIT	-0.003929	-0.039747	0.999473	-0.001044	-0.029415	-0.007921	0.010459	-0.04189
PASS	1.005452	-0.045203	-0.005399	0.001724	0.001836	0.002844	0.015173	-0.000318
RESIT	0.018257	0.024341	-0.019321	-0.049255	-0.043334	-0.047961	0.336244	0.775362
RESIT	-0.016895	0.190854	0.006798	-0.053634	-0.035932	-0.008747	0.011016	0.776829
DISTIN	-0.013222	-0.002516	0.037742	0.928493	0.038378	-0.050062	-0.021038	-0.017418

Table B45 Excerpt of 8 class blind testing result

The classification accuracy for the 8 class blind testing set was 93%, this is considered the minimum level of accuracy that is acceptable. Table B45 contains an excerpt of this data.

2.8 Results of 9 class experiment

Presented below are results from the 9 class training set. The 9 class training set trained poorly with erratic learning curves. This suggests some significant learning difficulty.

2.8.1 Confusion matrix 9 class T set

T	Redo	Resit	Compensate	Pass	Fail	Merit	Distinction	Commendation	Excellence
Redo	58.5	0	0	0	41.6	0	0	0	0
Resit	0	88.4	0	0	11.1	0	0	0	0
Compensate	0	0	40	60	0	0	0	0	0
Pass	0	0	0	100	0	0	0	0	0
Fail	0	18.1	0	0	81.8	0	0	0	0
Merit	0	0	0	7.1	0	85.7	7.1	0	0
Distinction	0	0	0	0	0	0	100	0	0
Commendation	0	0	0	0	0	0	17.2	59.5	23.2
Excellence	0	0	0	0	0	0	0	0	100

Table B46 T value confusion matrix result for 9 class set

As can be seen in table B46, the T value confusion matrix shows a confused set of classifications suggesting significant difficulty in learning.

2.8.2 Confusion matrix 9 class CV set

CV	Redo	Resit	Compensate	Pass	Fail	Merit	Distinction	Commendation	Excellence
Redo	0	0	0	0	100	0	0	0	0
Resit	0	77.7	0	0	33.3	0	0	0	0
Compensate	0	0.1	30.7	0	60.2	0	0	0	0
Pass	0	0	0	100	0	0	0	0	0
Fail	0	9.1	0	0	90.9	0	0	0	0
Merit	0	0	0	0	66.6	21.4	11.9	0	0
Distinction	0	0	0	0	0	0	100	0	0
Commendation	0	0	0	0	0	0	36.6	50.2	13.1
Excellence	0	0	0	0	0	0	0	0	100

Table B47 CV value confusion matrix result for 9 class set

As can be seen in table B47, the CV value confusion matrix shows poor learning result. The differences between the B46 and B47 suggest a poor ability to generalise.

2.8.3 Results of 9 class blind testing set

Desired	Out PASS	Out FAIL	Out MERIT	Out DISTIN	Out COMM	Out EXCEL	Out COMP	Out RESIT	Out REDO
PASS	0.648801	-0.01949	0.103907	-0.01883	-0.0348	-0.03602	0.295717	0.051408	-0.05067
MERIT	0.120409	0.043762	0.741339	0.065684	-0.00149	-0.04611	0.127839	-0.01662	-0.05123
FAIL	0.02247	0.049648	0.148884	0.088858	0.534072	0.347952	0.069326	-0.04647	0.046921
PASS	0.654667	-0.02439	0.11023	-0.00639	-0.03732	-0.03494	0.230393	0.039467	-0.05018
MERIT	0.265728	0.011756	0.644642	0.027398	-0.0112	-0.03749	0.029425	-0.03423	-0.04504
MERIT	0.098134	0.026145	0.750045	0.144708	-0.01214	-0.04458	0.053686	-0.02087	-0.0501
DISTIN	-0.02235	-0.00209	0.113174	0.598199	0.198463	0.190006	0.012456	-0.03191	-0.01437
PASS	0.654643	-0.0244	0.110234	-0.00635	-0.03733	-0.03493	0.230198	0.039447	-0.05018
COMP	0.648814	-0.01949	0.103912	-0.01883	-0.0348	-0.03602	0.295705	0.0514	-0.05067
REDO	-0.010842	0.327747	0.008163	-0.0411	0.003108	0.023627	-0.05386	-0.00179	0.668151

Table B48 Excerpt of 9 class blind testing result

The classification accuracy for the 9 class blind testing set was 82%, this is well below the minimum level of accuracy that is acceptable. Table B48 contains an excerpt of this data.

2.9 Results of 10 class experiment

Presented below are results from the 10 class training set. The 10 class training set exhibited severely erratic learning curves. This suggests some significant learning difficulty and some unusual confusion matrix results.

2.9.1 Confusion matrix 10 class T set

T	Repeat year	Redo	Resit	Compensate	Pass	Fail	Merit	Distinction	Commendation	Excellence
Repeat year	72.7	0	0	0	0	27.2	0	0	0	0
Redo	100	0	0	0	0	0	0	0	0	0
Resit	0	0	88.8	0	0	11.1	0	0	0	0
Compensate	0	0	9.1	80.9	10	0	0	0	0	0
Pass	0	0	0	0	100	0	0	0	0	0
Fail	0	0	0	0	0	100	0	0	0	0
Merit	0	0	0	0	0	0	100	0	0	0
Distinction	0	0	0	0	0	0	0	60	40	0
Commendation	0	0	0	0	0	0	0	0	100	0
Excellence	0	0	0	0	0	0	0	0	0	100

Table B49 T value confusion matrix result for 10 class set

As can be seen in table B49, the T value confusion matrix shows a reasonably positive learning result. However when compared with B50, the CV confusion matrix, more is revealed.

2.9.2 Confusion matrix 10 class CV set

CV	Repeat year	Redo	Resit	Compensate	Pass	Fail	Merit	Distinction	Commendation	Excellence
Repeat year	90.9	0	0	0	0	9.09	0	0	0	0
Redo	100	0	0	0	0	0	0	0	0	0
Resit	44.4	0	11.1	44.4	0	0	0	0	0	0
Compensate	0	0	0	25	50	25	0	0	0	0
Pass	0	0	0	0	81.2	0	18.7	0	0	0
Fail	71.4	0	0	0	0	28.5	0	0	0	0
Merit	0	0	0	0	0	0	100	0	0	0
Distinction	0	0	0	0	0	0	40	40	20	0
Commendation	0	0	0	0	0	0	0	11.1	88.8	0
Excellence	0	0	0	0	0	0	0	0	0	100

Table B50 CV value confusion matrix result for 2 class set

Table B50 shows the CV confusion matrix for the 10 class problem, this shows a poor ability to generalise. Tables B49 and B50 show very different results, B49 is moderately positive whereas B50 is very poor. The true accuracy of the network was determined with the blind test.

2.9.3 Result of 10 class blind testing set

Desired	Out PASS	Out FAIL	Out MERIT	Out DISTIN	Out COMM	Out EXCEL	Out COMP	Out RESIT	Out REDO	Out REPEAT
MERIT	0.015679	0.02184	0.858732	0.135101	0.009958	0.002437	0.002468	-0.00347	0.011871	-0.02728
PASS	0.913517	-0.0109	0.034158	-0.01967	-0.03498	-0.0036	0.03839	-0.01034	-0.00526	-0.04544
PASS	0.895654	-0.00503	0.014959	-0.02248	-0.03726	-0.00864	0.036905	-0.00538	-0.0061	-0.0449
RESIT	-0.03351	0.373729	-0.04796	-0.02287	-0.01731	-0.00918	-0.01383	0.018146	0.082691	0.158243
MERIT	0.015679	0.02184	0.858733	0.135102	0.009958	0.002437	0.002469	-0.00347	0.011872	-0.02728
FAIL	-0.02911	0.438832	-0.04596	-0.02535	-0.02177	-0.01039	-0.02691	-0.01224	0.168268	0.240041
COMM	-0.04588	0.006944	0.00913	0.196342	0.593068	0.191882	0.005544	0.008511	-0.046	-0.01033
MERIT	0.01539	0.021672	0.857209	0.134616	0.010137	0.002432	0.002214	-0.00371	0.011561	-0.02727
DISTIN	-0.04588	0.006948	0.009145	0.196353	0.593051	0.191878	0.005539	0.00851	-0.046	-0.01033
REDO	0.015679	0.02184	0.858733	0.135102	0.009958	0.002437	0.002469	-0.00347	0.011871	-0.02728

Table B51 Excerpt of the 10 class blind testing set

The classification accuracy for the 10 class blind testing set was 75%, this is well below the minimum level of accuracy that is acceptable. Table B51 contains an excerpt of this data.

3.0 Full results of the Noise experiment

Contained in this section are the full results for the noise experiments in chapter 5. The results are presented sequentially, detailing full T and CV confusion matrixes and excerpts of blind testing results.

3.1 Results of 0% noise experiment

Below the results of the 0% noise training set is presented. The 0% noise set trained very well and showed no difficulty in learning.

3.1.1 Confusion matrix 0% noise T set

T	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	0	96.4	3.5	0
Merit	0	3.4	92.1	3.4
Distinction	0	0	0	100

Table B52 T value confusion matrix result for 0% noise

As can be seen in table B52, the T value confusion matrix shows a positive learning result.

3.1.2 Confusion matrix 0% noise CV set

CV	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	18.7	81.2	0	0
Merit	0	10	90	0
Distinction	0	0	0	100

Table B53 CV value confusion matrix result for 0% noise

As can be seen in table B53, the CV value confusion matrix shows a very positive learning result, this is similar to table B52.

3.1.3 Result of 0% noise blind testing set

Des Fail	Des Pass	Des Merit	Des Distin	Out Fail	Out Pass	Out Merit	Out Distin
0	0	0	1	0.001681	-0.00431	-0.01986	0.985762
0	0	0	1	0.00554	-0.00354	-0.02095	0.987102
0	1	0	0	0.004072	0.985323	0.026075	-0.04755
0	0	1	0	-0.0454	0.175939	0.660875	0.058791
1	0	0	0	0.943588	0.060838	-0.0014	-0.02409
0	0	0	1	-0.01573	0.002796	-0.00438	0.955757
0	1	0	0	0.008226	0.991209	0.019671	-0.04826
0	1	0	0	0.003284	0.964382	0.039099	-0.04652
0	0	0	1	0.005533	-0.00358	-0.02097	0.98712
0	1	0	0	0.01038	0.972605	0.020595	-0.04854

Table B54 Excerpt of the 0% blind testing set

The classification accuracy for the 0% noise blind testing set was 97%, this is well within the acceptable error margin. See table B54 for an except of this testing data

3.2 Results of 5% noise experiment

Below the results of the 5% noise training set is presented. The 5% noise set trained very well and showed no difficulty in learning. In fact the results show an increase in classification accuracy with 5% noise.

3.2.1 Confusion matrix 5% noise T set

T	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	3.9	96.1	0	0
Merit	0	0	95.6	4.3
Distinction	0	0	0	100

Table B55 T value confusion matrix result for 5% noise

As can be seen in table B55, the T value confusion matrix shows a positive learning result.

3.2.2 Confusion matrix 5% noise CV set

CV	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	0	100	0	0
Merit	0	10	90	0
Distinction	0	0	0	100

Table B56 CV value confusion matrix results for 5% noise

As can be seen in table B56, the CV value confusion matrix shows a very positive learning result, this is similar to table B55.

3.2.3 Result of 5% noise blind testing set

Des Fail	Des Pass	Des Merit	Des Distin	Out Fail	Out Pass	Out Merit	Out Distin
0	0	0	1	0.012615	-0.00884	0.024408	0.959908
0	0	0	1	0.012615	-0.00884	0.024408	0.959908
0	1	0	0	0.013642	0.960711	0.030729	0.003028
0	0	1	0	-0.04998	0.073785	0.912014	0.02006
1	0	0	0	0.963716	0.042243	-0.00866	0.007934
0	0	0	1	0.012562	-0.00883	0.02446	0.95988
0	1	0	0	0.013906	0.961221	0.030013	0.003248
0	1	0	0	0.013774	0.960969	0.030368	0.003139
0	0	0	1	0.012615	-0.00884	0.024408	0.959908
0	1	0	0	0.013904	0.961221	0.030014	0.003248

Table B57 Excerpt of the 5% blind testing set

The classification accuracy for the 5% noise blind testing set was 98%, this is well within the acceptable error margin. See table B57 for an except of this testing data.

When compared with 0% noise, table 54, the 5% noise shows an increase in accuracy. Although counterintuitive, this result is not unexpected, noise has been shown to be beneficial to learning in neural networks Sietsma and Dow (1991)

3.3 Result of the 10% noise experiment

Below the results of the 10% noise training set is presented. The 10% noise set trained very well and showed no difficulty in learning.

3.3.1 Confusion matrix 10% noise T set

T	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	0	96.4	3.5	0
Merit	0	4.3	91.3	4.3
Distinction	0	0	0	100

Table B58 T value confusion matrix result for 10% noise

As can be seen in table B58, the T value confusion matrix shows a positive learning result.

3.3.2 Confusion matrix 10% noise CV set

CV	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	27.2	63.3	9	0
Merit	0	20	60	20
Distinction	0	0	40	60

Table B59 CV value confusion matrix result for 10% noise

As can be seen in table B59, the CV value confusion matrix shows a very positive learning result, this is similar to table B58.

3.3.3 Result of 10% noise blind testing set

Des Fail	Des Pass	Des Merit	Des Distin	Out Fail	Out Pass	Out Merit	Out Distin
0	0	0	1	0.013023	0.001135	0.119032	0.850674
0	0	0	1	0.013023	0.001135	0.119032	0.850674
0	1	0	0	0.02108	0.879168	0.102193	-0.02852
0	0	1	0	-0.00064	0.837609	0.162668	-0.02555
1	0	0	0	0.906964	0.088113	0.010795	-0.00062
0	0	0	1	0.012997	0.001137	0.119151	0.850608
0	1	0	0	0.021105	0.879175	0.10217	-0.02852
0	1	0	0	0.02047	0.879017	0.102805	-0.02855
0	0	0	1	0.013023	0.001135	0.119032	0.850674
0	1	0	0	0.0211	0.879175	0.102174	-0.02852

Table B60 Excerpt of the 10% blind testing set

The classification accuracy for the 10% noise blind testing set was 97%, this is well within the acceptable error margin, table B60 contains an except of this testing data.

3.4 Result of the 15% noise experiment

Below the results of the 15% noise training set is presented. The 15% noise set trained very well and showed no difficulty in learning.

3.4.1 Confusion matrix 15% noise T set

T	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	0	96.4	3.5	0
Merit	0	4.3	91.3	4.3
Distinction	0	0	0	100

Table B61 T value confusion matrix result for 15% noise

As can be seen in table B61, the T value confusion matrix shows a positive learning result.

3.4.2 Confusion matrix 15% noise CV set

CV	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	18.6	81.3	0	0
Merit	0	0	100	0
Distinction	0	0	0	100

Table B62 CV value confusion matrix result for 15% noise

As can be seen in table B62, the CV value confusion matrix shows a very positive learning result, this is similar to table B61.

3.4.3 Result of 15% noise blind testing set

Des Fail	Des Pass	Des Merit	Des Distin	Out Fail	Out Pass	Out Merit	Out Distin
0	0	0	1	0.00066	0.012581	0.166752	0.814255
0	0	0	1	0.00066	0.012581	0.166752	0.814255
0	1	0	0	0.163188	0.799162	0.12963	0.072857
0	0	1	0	0.049328	0.555127	0.251362	0.027006
1	0	0	0	0.857825	0.099532	-0.00479	-0.00268
0	0	0	1	0.000354	0.012069	0.168365	0.813727
0	1	0	0	0.164504	0.800634	0.12888	0.073265
0	1	0	0	0.164235	0.800335	0.129033	0.073182
0	0	0	1	0.00066	0.012581	0.166752	0.814255
0	1	0	0	0.164503	0.800632	0.128881	0.073265

Table B63 Excerpt of the 15% blind testing set

The classification accuracy for the 15% noise blind testing set was 97%, this is well within the acceptable error margin, table B63 contains an except of this testing data.

3.5 Result of the 20% noise experiment

Below the results of the 20% noise training set is presented. The 20% noise showed some difficulty in learning through erratic learning curves.

3.5.1 Confusion matrix 20% noise T set

T	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	14.2	75	10.7	0
Merit	0	4.3	91.3	4.3
Distinction	0	0	0	100

Table B64 T value confusion matrix result for 20% noise

As can be seen in table B64, the T value confusion matrix shows a moderately positive learning result. There is some confusion present, especially between Merit, Distinction and Pass.

3.5.2 Confusion matrix 20% noise CV set

CV	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	18.1	63.3	18.1	0
Merit	0	20	40	40
Distinction	0	0	0	100

Table B65 CV value confusion matrix result for 20% noise

As can be seen in table B65, the CV value confusion matrix shows a moderately negative learning result. As with table B64, confusion exists with Merit and Pass. This suggests difficulty in generalising to unseen examples.

3.5.3 Result of 20% noise blind testing set

Des Fail	Des Pass	Des Merit	Des Distin	Out Fail	Out Pass	Out Merit	Out Distin
0	0	0	1	-0.0092	0.015097	0.2885	0.696348
0	0	0	1	-0.0092	0.015097	0.2885	0.696348
0	1	0	0	0.053051	0.832258	0.143902	-0.03054
0	0	1	0	0.05004	0.835867	0.14636	-0.0303
1	0	0	0	0.815679	0.171072	-0.00459	-0.00341
0	0	0	1	-0.0092	0.015097	0.288509	0.696329
0	1	0	0	0.05404	0.831034	0.14309	-0.03061
0	1	0	0	0.053575	0.831613	0.143479	-0.03058
0	0	0	1	-0.0092	0.015097	0.2885	0.696348
0	1	0	0	0.053587	0.831599	0.14347	-0.03058

Table B66 Excerpt of the 20% blind testing set

The classification accuracy for the 20% noise blind testing set was 91%, this is well below the minimum level of acceptable accuracy, table B66 contains an except of this testing data.

3.6 Result of the 25% noise experiment

Below the results of the 25% noise training set is presented. The 25% noise showed severe difficulty in learning, displaying erratic learning curves.

3.6.1 Confusion matrix 25% noise T set

T	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	0	96.4	3.5	0
Merit	0	4.3	91.3	4.3
Distinction	0	0	4.5	95.4

Table B67 T value confusion matrix result for 25% noise

As can be seen in table B67, the T value confusion matrix shows a moderately positive learning result. There is some confusion present, especially between Merit, Distinction and Pass.

3.6.2 Confusion matrix 25% noise CV set

CV	Fail	Pass	Merit	Distinction
Fail	100	0	0	0
Pass	27.2	63.6	9	0
Merit	0	10	60	30
Distinction	0	0	30	70

Table B68 CV value confusion matrix result for 25% noise

As can be seen in table B68, the CV value confusion matrix shows a negative learning result. As with table B67, confusion exists with Pass, Merit and Distinction. This suggests severe difficulty in generalising to unseen examples.

3.6.3 Result of 25% noise blind testing set

Des Fail	Des Pass	Des Merit	Des Distin	Out Fail	Out Pass	Out Merit	Out Distin
0	0	0	1	-0.00978	-0.04964	0.13962	0.670239
0	0	0	1	-0.00978	-0.04964	0.13962	0.67024
0	0	1	0	0.072988	0.952179	0.11343	-0.0498
0	0	1	0	0.033682	0.904764	0.155374	-0.0502
1	0	0	0	0.726213	0.478474	-0.054	-0.05515
0	0	0	1	-0.00927	-0.04966	0.141004	0.667935
0	1	0	0	0.408279	0.753249	-0.01288	-0.05156
0	1	0	0	0.033598	0.90342	0.153946	-0.05027
0	0	0	1	-0.00959	-0.04965	0.140146	0.669393
0	1	0	0	0.033598	0.903418	0.153945	-0.05027

Table B69 Excerpt of the 25% blind testing set

The classification accuracy for the 25% noise blind testing set was 89%, this is well below the minimum level of acceptable accuracy, table B66 contains an except of this testing data.

Appendix C

Instructions

Dear participant,

Thank you for agreeing to co-operate in this spreadsheet development experiment. Your input is greatly appreciated and will contribute to research by the Cardiff school of management.

Instructions for the experiment

1. Complete questionnaire 1
2. Next turn over the page to find task 1
3. Read the instructions at the top of each sheet and complete the task as best you can
4. Turn over the page to the next task and complete, repeat until you have finished all tasks.
5. Once you have finished all the tasks, complete questionnaire 2
6. The experiment has finished, thank you for your co operation.

Questionnaire 1

Please tick as appropriate

Question 1.1

Which category of age do you fall in?

18 -25	25 – 30	30 – 35	35 – 40	41 or above

Question 1.2

Please indicate your Gender

Male	Female

Question 1.3

How would you rate yourself as a spreadsheet user?

No experience	Novice	Competent	Experienced	Very experienced

Question 1.4

How many years have you been using spreadsheets?

Never used them before	Under 1 year	1-5 years	5 to 9 years	10 years or more

Question 1.5

What training have you had in spreadsheets?

None	Self Taught	Undergraduate education	Post graduate education	In house training from employers	Private training company	Other (please specify)

Give examples of percentage marks for coursework and exam and then give resulting grade for the criteria below
Give 2 examples of grade (pass and fail) where a Pass is achieved if the exam mark is equal to or greater than 40

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Name**	**Course**	**Exam**	**Grade**		**Criteria**						
2	Bert	Electronics				Pass	>=40	Grade is pass if exam > 40				
3	George	Biology				Fail	<40					
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Task 1

Give examples of percentage marks for coursework and exam and then give resulting grade (pass or fail) for the criteria below provide 2 examples of each grade, Pass is achieved if the average of coursework and exam are greater than or equal to 40

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Name**	**Course**	**Coursework**	**Exam**	**Grade**		**Criteria**					
2	Bert	Electronics					Pass	>=40	Grade is pass if average of marks are a pass			
3	George	Biology					Fail	<40				
4	Luey	Chemistry										
5	Tia	Computing										
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Task 2

Give examples of percentage marks for coursework and exam and then give resulting grade for the criteria below
Give 2 examples of each grade (pass and fail) where a pass grade is achieved if <u>both</u> coursework and exam are greater than or equal to 40

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Name**	**Course**	**Coursework**	**Exam**	**Grade**		**Criteria**						
2	Bert	Electronics					Pass	>=40	Grade is pass if both marks are pass				
3	George	Biology					Fail	< 40					
4	Luey	Chemistry											
5	Tia	Computing											
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													

Task 3

Give examples of percentage marks for coursework and exam and then give resulting grade for the criteria below
Give 2 examples of each grade (pass, merit, distinction, fail) refer to the criteria for grade boundaries

	A	B	C	D	E	F	G	H	I	J	K
1	**Name**	**Course**	**Coursework**	**Exam**	**Grade**		**Criteria**				
2	Bert	Electronics					Pass	>=40	Grade is pass if average marks are pass		
3	George	Biology					Merit	>55	Average mark must be Merit		
4	Luey	Chemistry					Distinction	>70	Average mark must be Distinction		
5	Tia	Computing					Fail	<40			
6	Stanley	Maths									
7	Dell	Geography									
8	Tanja	History									
9	Kelly	French									
10											
11											
12											
13											
14											
15											
16											

Task 4

Give examples of percentage marks for coursework and exam and then give resulting grade for the criteria below
Give 2 examples of each grade (pass, merit, distinction, fail) refer to the criteria for grade boundaries

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Name**	**Course**	**Coursework**	**Exam**	**Grade**		**Criteria**						
2	Bert	Electronics					Pass	>=40	Grade is pass if both marks are pass				
3	George	Biology					Merit	>55	Both marks must be Merit				
4	Luey	Chemistry					Distinction	>70	Both marks must be Distinction				
5	Tia	Computing					Fail	<40					
6	Stanley	Maths											
7	Dell	Geography											
8	Tanja	History											
9	Kelly	French											
10													
11													
12													
13													
14													
15													
16													

Task 5

Once task 5 is completed please fill out questionnaire 2 overleaf.

Questionnaire 2

Please fill in this questionnaire by ticking **only one** appropriate box.

Question 2.1

a) Did you understand the instructions for task 1?

No	Don't know	Yes

b) Did you successfully complete task 1?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 1?

Very easy	Easy	Average	Hard	Very Hard

Question 2.2

a) Did you understand the instructions for task 2?

No	Don't know	Yes

b) Did you successfully complete task 2?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 2?

Very easy	Easy	Average	Hard	Very Hard

Question 2.3

a) Did you understand the instructions for task 3?

No	Don't know	Yes

b) Did you successfully complete task 3?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 3?

Very easy	Easy	Average	Hard	Very Hard

Question 2.4

a) Did you understand the instructions for task 4?

No	Don't know	Yes

b) Did you successfully complete task 4?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 4?

Very easy	Easy	Average	Hard	Very Hard

Question 2.5

a) Did you understand the instructions for task 5?

No	Don't know	Yes

b) Did you successfully complete task 5?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 5?

Very easy	Easy	Average	Hard	Very Hard

End of test

Appendix D

Questionnaire 1

Please tick as appropriate

Question 1.1

Which category of age do you fall in?

18 -25	25 – 30	30 – 35	35 – 40	41 or above

Question 1.2

Please indicate your Gender

Male	Female

Question 1.3

How would you rate yourself as a spreadsheet user?

No experience	Novice	Competent	Experienced	Very experienced

Question 1.4

How many years have you been using spreadsheets?

Never used them before	Under 1 year	1-5 years	5 to 9 years	10 years or more

Question 1.5

What training have you had in spreadsheets?

None	Self Taught	Undergraduate education	Post graduate education	In house training from employers	Private training company	Other (please specify)

Create a formula that will output either pass or fail in the 'Grade' column for the spreadsheet below.
Grade is calculated as Pass if the exam mark is greater than or equal to 40

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Name**	**Course**	**Exam**	**Grade**		**Criteria**						
2	Brian	Mechanics	44			Pass	>=40	Grade is pass if exam >=40				
3						Fail	<40					
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Task 1

Create a formula that will output either: Pass or Fail
Grade is calculated as an <u>average</u> of exam and coursework, Pass is achieved if the average mark is a pass

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Name**	**Course**	**Coursework**	**Exam**	**Grade**		**Criteria**					
2	Carol	Electronics	78	67			Pass	>=40	Grade is pass if average marks are pass			
3							Fail	<40				
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Task 2

Create a formula that will output either: Pass or Fail
In this example, to achieve a pass <u>both</u> marks must be >40

	A	B	C	D	E	F	G	H	I	J
1	**Name**	**Course**	**Coursework**	**Exam**	**Grade**		**Criteria**			
2	Bert	Computing	43	56			Pass >=	40	Grade is pass if both marks are pass	
3							Fail <	40		
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										

Task 3

Create a formula that will output Fail, Pass, Merit or Distinction in the grade column for the spreadsheet below
You have to write a formula that will produce a grade based on an <u>average</u> of coursework and exam according to the criteria

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Name**	**Course**	**Coursework**	**Exam**	**Grade**		**Criteria**					
2	Jenny	HRM	50	80			Pass	>=40	Average mark must be Pass			
3	Mike	Marine biology	40	67			Merit	>55	Average mark must be Merit			
4	John	Sport science	56	65			Distinction	>70	Average mark must be Distinction			
5	Paul	Music	25	43			Fail	<40				
6	George	Politics	54	78								
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Task 4

Create a formula that will output Fail, Pass, Merit or Distinction in the grade column for the spreadsheet below
You have to write a formula that will produce a grade based on an average of coursework and exam according to the criteria

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Name**	**Course**	**Coursework**	**Exam**	**Grade**		**Criteria**					
2	Dave	Chemistry	35	45			Pass	>=40	Grade is pass if both marks are pass			
3	Simon	Economics	40	39			Merit	>55	Both marks must be Merit			
4	Nigel	Business	41	56			Distinction	>70	Both marks must be Distinction			
5	Ann	Finance	48	34			Fail	<40				
6	Karen	Customer relations	60	88								
7	Alex	Psychology	36	66								
8												
9												
10												
11												
12												
13												
14												
15												
16												

Task 5

Once task 5 is completed please complete questionnaire 2 overleaf

Questionnaire 2

Please fill in this questionnaire by ticking **only one** appropriate box.

Question 2.1

a) Did you understand the instructions for task 1?

No	Don't know	Yes

b) Did you successfully complete task 1?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 1?

Very easy	Easy	Average	Hard	Very Hard

Question 2.2

a) Did you understand the instructions for task 2?

No	Don't know	Yes

b) Did you successfully complete task 2?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 2?

Very easy	Easy	Average	Hard	Very Hard

Question 2.3

a) Did you understand the instructions for task 3?

No	Don't know	Yes

b) Did you successfully complete task 3?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 3?

Very easy	Easy	Average	Hard	Very Hard

Question 2.4

a) Did you understand the instructions for task 4?

No	Don't know	Yes

b) Did you successfully complete task 4?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 4?

Very easy	Easy	Average	Hard	Very Hard

Question 2.5

a) Did you understand the instructions for task 5?

No	Don't know	Yes

b) Did you successfully complete task 5?

No	Probably not	Don't know	Probably	Yes

c) How difficult was task 5?

Very easy	Easy	Average	Hard	Very Hard

End of test