Internet Banking Fraud Detection Using Prudent Analysis

Oarabile Omaru Maruatona

This thesis is submitted in total fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Science, Information Technology and Engineering (SITE)

University of Ballarat

P O Box 663

University Drive, Mt Helen

Ballarat, VIC 3353, Australia

Supervisors

Dr. Peter Vamplew

Dr. Richard Dazeley

February 2013

Abstract

The threat posed by cybercrime to individuals, banks and other online financial service providers is real and serious. Through phishing, unsuspecting victims' Internet banking usernames and passwords are stolen and their accounts robbed. In addressing this issue, commercial banks and other financial institutions use a generically similar approach in their Internet banking fraud detection systems. This common approach involves the use of a rule-based system combined with an Artificial Neural Network (ANN).

The approach used by commercial banks has limitations that affect their efficiency in curbing new fraudulent transactions. Firstly, the banks' security systems are focused on preventing unauthorized entry and have no way of conclusively detecting an imposter using stolen credentials. Also, updating these systems is slow and their maintenance is labour-intensive and ultimately costly to the business. A major limitation of these rule-bases is brittleness; an inability to recognise the limits of their knowledge.

To address the limitations highlighted above, this thesis proposes, develops and evaluates a new system for use in Internet banking fraud detection using Prudence Analysis, a technique through which a system can detect when its knowledge is insufficient for a given case. Specifically, the thesis proposes the following contributions:

- Conduct comprehensive comparisons of two successful prudence methods: Rated MCRDR (RM) and Ripple Down Models (RDM).
- Redevelopment of Multiple Classifications RDM from Single Classification RDM.
- Development of a new prudence method Integrated Prudence Analysis (IPA) by combining RM and RDM.
- Introduction and application of RDR prudence to Internet banking fraud detection.

Statement of Authorship

Except where explicit reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis by which I have qualified for or been awarded another degree or diploma. No other person's work has been relied upon or used without due acknowledgment in the main text and bibliography of the thesis.

Oarabile Omaru Maruatona

SIGNED: 

Publications by Author

The following is a list of academic, peer reviewed publications produced in the course of this project.

- Maruatona, O.O, Vamplew, P; Dazeley, R. RM and RDM, a Preliminary Evaluation of two Prudent RDR Techniques. The 6th Pacific Knowledge Acquisition Workshop (PKAW 2012). Kuching, Malaysia.

The preliminary results published at the PKAW conference were the first phase of comparisons between RM and RDM. Complete evaluation results are presented in chapter six.

- Maruatona, O.O, Vamplew, P; Dazeley, R. Prudent Fraud Detection in Internet Banking. Cybercrime and Trustworthy Computing Workshop (CTC 2012). Ballarat, Australia.

The CTC publication discussed the viability of prudence in fraud detection components of Internet banking systems. The paper presented some early test results using the IPA system.

Acknowledgements

Kwalo ya thesis e, ke karolo ya bobedi mo mosepeleng wa me wa PhD. Thesis e e simolotse dikgwedinyana morago ga ke sena go digela thoto e nngwe ya mmereko wa Programming e mo go one go dirisiwang compitara go dira tiro nngwe e neng e ka se kgonege ka diatla kgotsa e le bonya ga e dirwa ke motho. Tshwetso ya me go dira PhD ga e a tsalwa fela ke gore ne ke batla go nna le yone PhD. Ke tshwetso e e akareditseng dilo tse dintsi e bile ke e rerisitse ba lwapa la me. Ke itumelela gore botlhe ba ke ba rerisitseng ba mphile tletla ya go dira se pelo ya me e neng e se eleditse. Ka bokhutswhane nka re mosepele wame wa PhD e nnile nngwe ya dikgwetlho tse di tona tse ke di ineileng ebile ke motlotlo go bo ke ithutile tse ke di ithutileng, go bo ke bone mafelo a ke a boneng, go bo ke kopane le be ke kopnaneng nabo, le tse di ngwe tse di molemo tse ke di jeleng ka mokgwa mongwe ke di lebogela fela thata. Ke leboga thata Thembi Omaru, mme yo ke fudugetseng mo toropong ya Ballarat le ene ga ke tla go simolola mosepele o. Re ithutile thata, tota re amogane gole go ntsi nna le Thembi. Ke nnile lesego thata go thulana le ene ka nako e rotlhe re boneng mosola wa go simolola botsala jwa rona. Ke rata go leboga go menagane botlhe ba ba tsereng karolo mo ponagatsong ya phenyo e. Ke leboga mogolwane wa me wa tsa patisiso, Dr Peter Vamplew, mogolwane wa bobedi wa tsa patisio, Dr Richard Dazeley. Malebo gape go moeteledi was lefelo la patisiso la Internet Commerce Security Laboratory, Associate Professor Paul Watters. Malebo a magolo go dikompone tse di thusang ICSL ka tsa madi le go fa tiro ya rona boleng jwa tsa kgwebo, Westpac Bank Ltd, IBM Australia le Australian Federal Police.

This thesis is a result of a collective effort of a number of people. It was Mary Imam who suggested that I apply for a PhD when I was looking for jobs and Masters Scholarships in Australia. Thanks Mary for your steady support throughout the years.

I would like to acknowledge my supervisors Dr Peter Vamplew and Dr Richard Dazeley, who were pivotal in getting this project to where it is now. I have realised how fortunate I was to have the two of them as supervisors and am grateful for the collegial working relationship we had over the last three years.

I would also like to thank the Internet Commerce Security Laboratory (ICSL) and its partners Westpac Bank Ltd, IBM Australia and the Australia Federal Police for the generous support

and for the opportunity to have undertaken my PhD under their auspices. It was through them that my PhD earned the practical and commercial relevance it now possesses. Special thanks to colleagues at the ICSL and our industry contributors from Westpac, IBM and the AFP. Your insightful feedback was much appreciated.

To the University of Ballarat; thanks for the opportunity. Every day was worth it. And to all members of staff at the Research Services Office; thank you for everything.

Lastly, I would like to extend my heartfelt thanks to my family for standing by me in all I do and my wife Thembi, who was always there when I needed her. Your support inspires me and this PhD was accomplished with your unwavering support.

To Thembi Omaru. Your love and support has always been my strength. Ngiya bonga sana.

Contents

1. Introduction 1

1.1 The State of Affairs 1

1.2 Current Approaches to Internet Banking Fraud Detection 1

1.3 Limitations of Current Fraud Detection Methods 2

1.4 Prudent Ripple Down Rules: a viable Solution to Fraud Detection 3

1.5 Project Plan and Contributions 4

1.6 Thesis overview 5

2. Literature Review: Fraud Detection 9

2.1 Introduction 9

2.2 Internet Fraud 9

2.3 Online Banking, a Victim of Internet Fraud 11

2.4 Detecting Fraud with Outlier Detection 13

2.5 Applying Outlier Detection in Intrusion Detection 15

2.6 Outlier Detection in Fraud Detection 19

2.7 Remarks and Observations 24

2.8 Chapter Summary 24

3. Literature Review: Knowledge Based Systems and Ripple Down Rules 26

3.1 Introduction 26

3.2 Knowledge-based Systems 26

3.3 Rule-Based Systems 28

3.4 Ripple Down Rules 31

3.5 RDR Learning 33

3.6 Multiple Classifications RDR 35

3.7 MCRDR Learning 37

3.8 Prudence in Knowledge Bases 41

3.9 Chapter Summary 43

4. Prudent RDR methods 45
4.1 Introduction 45
4.2 Rated MCRDR 45
4.3 Ripple Down Models 55
4.4 MCRDR based RDM 63
4.5 Integrated Prudence Analysis 64
4.6 Chapter Summary 67

5. Methodology 68
5.1 Introduction 68
5.2 The Need for KBS evaluation 68
5.3 Evaluation Metrics for KBS 69
5.4 Evaluation in RDR KBS 76
5.5 Creating a Simulated Expert 80
5.6 Evaluation Metrics 82
5.7 Datasets 87
5.8 Chapter Summary 89

6. Results on Public Datasets 90
6.1 Introduction 90
6.2 Single and Multiple Classifications RDM 90
6.3 Simple Accuracy: RM versus RDM 92
6.4 Prudence Accuracy: RM versus RDM 93
6.5 Simple Accuracy Before and After Prudence 97
6.6 IPA Prudence Accuracy 99
6.7 Integrating Two Prudent Methods: Does It Work? 102
6.8 Chapter Summary 105

7. IPA Results on Internet Banking Data 106
7.1 Introduction 106
7.2 Internet Banking Fraud 106
7.3 Internet Banking Transactions 108
7.4 Obfuscation and Online Banking Data 110

7.5 IPA versus Commercial System 113
7.6 Internet Banking Fraud Detection Framework 115
7.7 Chapter Summary 116

8. Conclusion 118
8.1 Introduction 118
8.2 The Situation 118
8.3 The Project 120
8.4 The Results 120
8.5 Contributions 121
8.6 Conclusion 122

9. Bibliography 123
Appendix A: Acronyms 132

List of Figures

Figure 2-1. Some of the Falcon s main components (FICO, 2011) 21

Figure 3-1. Basic schematic of a KBS (Giarratano & Riley, 2005) 27

Figure 3-2. Components of a simple rule 29

Figure 3-3. Change in complexity of a rule over 3 years. 31

Figure 3-4. The RDR tree structure. 33

Figure 3-5. An MCRDR for X= {b, d, f, k, o, h, e, m, t, y} 37

Figure 3-6. The two types of Prudence and their examples 43

Figure 4-1. An overview of RM's main components. 46

Figure 4-2. A simple three layered ANN with input, hidden and output layers. 49

Figure 4-3. Dynamic addition of a new input node to an ANN. 53

Figure 4-4. Adding a new input and hidden node to an ANN 53

Figure 4-5. Main components of RDM.. 56

Figure 4-6. Output of RDR RDM inference engines 57

Figure 4-7. The IPA_{OR} /IPA_{AND} system. 65

Figure 4-8. A generic overview of the IPA_{ANN} system.. 66

Figure 5-1. FPA questions 70

Figure 5-2. ROC curves for 4 systems. 74

Figure 5-3. See5 decision tree and induction rules from the iris dataset. 81

List of Tables

Table 3-1. Attributes of two models of sports cars, the 16I and the 18I 34

Table 3-2. A sample difference list of two different editions of sports cars 34

Table 3-3. Early prudence system's performance statistics 42

Table 5-1. FPA assessment matrix 71

Table 5-2. MCRDR accuracy after learning from a See5 and induction rules SE. 82

Table 5-3. Description of used public datasets and SEs. 88

Table 5-4. Labwizard statistics for 3 knowledge bases 89

Table 6-1. Comparison of two SC-RDM and MC-RDM on prudence accuracy 91

Table 6-2. MCRDR's Simple accuracy in numerical datasets 92

Table 6-3. MCRDR's simple accuracy on categorical datasets 93

Table 6-4. Specificity, Sensitivity and BA of RM and RDM in numerical datasets 94

Table 6-5. RM and RDM's Specificity, Sensitivity and BA in categorical datasets 95

Table 6-6. Comparison of simple accuracy before and after prudence 98

Table 6-7. IPA prudence accuracy on numerical datasets 100

Table 6-8. IPA prudence accuracy statistics on the categorical datasets 101

Table 6-9. IPA, RM and RDM's BA in numerical datasets. 102

Table 6-10. RM, RDM and IPA Acc and BA on categorical data 103

Table 6-11. IPA's simple accuracy before and after prudence 104

Table 7-1. Description of online banking transaction attributes 109

Table 7-2. Demonstration of three obfuscation types 110

Table 7-3. IPA simple accuracy on online banking data 111

Table 7-4. IPA prudence accuracy on online banking data 112

Table 7-5. IPA vs. Commercial system detection rates 114

1. Introduction

1.1 The State of Affairs

Cybercrime in general continues to be a serious problem for individuals and businesses using or offering online services. According to published statistics, the financial and banking industry especially bears most of the effects of these crimes through Internet banking fraud, since the ultimate motive is usually financial gain (APWG, 2010). One of the most prevalent methods of Internet banking fraud is Phishing, which involves the use of technology and deceit to illicitly acquire an unsuspecting victim's credentials (APWG, 2011). These credentials (including personal details, online banking usernames and passwords) are then used by fraudsters to access a victim's emails, social security benefits and bank account, eventually resulting in the removal of funds from a victim's account. In 2008, phishing related Internet banking fraud was said to have cost banks around the world more than US$3 billion (McCombie, 2008). It has been reported also that beyond 2010, there would be a rise in sophisticated and more effective phishing techniques, ultimately resulting in more losses of funds through Internet banking fraud (RSA, 2010). This situation necessitates the development and wide deployment of more efficient and rapid Internet banking fraud detection systems.

1.2 Current Approaches to Internet Banking Fraud Detection

Online banking fraud detection systems which rely purely on user account and password authentication have no way of differentiating an account's legitimate user and a fraudster as long as the correct login credentials have been used to log in. This is one of the features enabling the prevalence of phishing, and also an area that security in Internet banking needs to focus on. The approach used by commercial banks to detect frauds in online transactions is generically similar. Reports and white papers of four commercially applied online payment fraud detection systems have shown that the common approach is the use of a Rule-base system combined with an Artificial Neural Network (ANN). The Falcon Fraud Manager, the Proactive Risk Manager (PRM) and the SAS Fraud Management system are examples of fraud detection systems used in commercial Internet banking systems (FICO, 2011; ACI Worldwide,

2011; SAS, 2007). These systems are extensively used around the world including in more than 40 countries, at 43000 sites and by half of the world's top 20 banks. Each of these fraud detection systems' architectures involves the use of a neural network to profile user behaviours and a conventional rule-base to create and maintain rules that define normal and anomalous behaviour (FICO, 2011; ACI Worldwide, 2011; SAS, 2007).

1.3 Limitations of Current Internet Banking Fraud Detection Methods

Most online banking fraud detection systems have no immediate way of realising when a fraudster logs in with correct but stolen details. Many of these systems will actually accept the correct username and password for a registered customer even if the user at the time is a fraudster using stolen credentials. Although two way authentications reduce the extent of damage posed by this flaw, there still remains the threat of mobile phone hackings, interceptions and viruses through which the fraudsters can evade dual authentications (Weber & Darbellay, 2010; Androulidakis & Papapetros, 2008). Instead of just preventing unauthorised entry, Internet banking fraud detection systems have to be able to detect a fraud immediately within a compromised account. The systems need to have some capability of conclusively determining whether a given user is the legitimate account holder or an impersonator.

Slow approach to Knowledge Acquisition

Conventional rule-bases used by commercial Internet banking systems have been criticised for a number of performance limiting inadequacies. Their approach to Knowledge Acquisition (KA) has been faulted for being too slow, labour intensive and costly for business (Richards, 2003). Also, maintenance in RDR is integrated with KA and not an additional task as in conventional rule-bases.

Brittle Rule-bases

These rule bases have also been labelled brittle; a phenomenon when Expert Systems attempt to use current knowledge to cover a case beyond this knowledge (Prayote & Compton, 2006). Such systems do not have a way of knowing when a new case cannot be processed using existing knowledge. They always attempt to give an answer even if it may be inaccurate.

Slow to adapt to new frauds

Consequently, a brittle Internet banking fraud detection system is less accurate since it always depends on its current knowledge even for novel cases where this knowledge is insufficient. It has also been consistently reported that modern fraud screening systems need capabilities for more accurate, automated and most importantly rapid detection of anomalous patterns (SAS, 2007; IBM, 2008). Rapid detection will enable fraud teams to respond to threats and potential threats in real time and will help save resources expended whenever a fraud has occurred.

1.4 Prudent Ripple Down Rules: a viable Solution to Internet Banking Fraud Detection

The Ripple Down Rules (RDR) approach to KA has been shown to have notable advantages over conventional rule-bases. RDR methods have been shown to have a better, faster and less costly rule addition and maintenance than conventional rule-bases (Kang, Compton, & Preston, 1995; Richards, 2003; Prayote, 2007).

No need for knowledge engineer

Conventional rule-bases typically require a knowledge engineer and a domain expert to build. Conversely, RDR knowledge bases only require a domain engineer. One of the earliest RDR systems, PIERS was described as user maintained and not requiring knowledge engineering expertise (Edwards, Compton, Malor, Srinivasan, & Lazarus, 1993). Similar RDR systems have since been successfully used in other applications including in web browsers, help desk systems, online shopping, email management systems and Network traffic classification (Kang, Compton, & Preston, 1995; Richards, 2003; Prayote, 2007).

Ability to realise limits

To address the issue of brittleness, RDR methods have an added feature called Prudence. Prudence allows the knowledge engine to realise when a current case is beyond the system's expertise. In such situations, the prudent system would issue a warning for the case to be investigated by the expert, resulting in the addition of knowledge to cover the case. In contrast, a brittle system would attempt to give a conclusion based on insufficient knowledge. The conclusion will most probably be erroneous and it would take some time for the system's administrators/experts to correct the error. For Internet banking, prudent fraud

detection systems mean accurate and rapid detection of new fraud patterns, saving time, human resources and money for both the financial institutions and their customers. Some of the theoretically and domain successful prudent RDR methods include Rated Multiple Classification Ripple Down Rules (RM) and Ripple Down Models (RDM) (Dazeley, 2007; Prayote, 2007). RM uses patterns in the MCRDR engine's inferencing paths to train an ANN to contribute to the system's understanding of a domain. RDM applies an outlier detection method to homogenized profiles from an RDR engine.

Impressive classification accuracy

Both RM and RDM have shown impressive classification and prediction ability in a number of public and proprietary datasets and have been commended for their performance (Dazeley, Park, & Kang, 2011; Prayote & Compton, 2006; Dazeley & Kang, 2008). The two systems' fundamental difference is that RM is structural and uses the inferencing engine's structure to deduce additional context about the domain. RDM is attribute-based and exploits the accuracy of outlier detection methods in homogenous profiles. To date, there have not been focused and direct comparisons of the two methods and to establish which domain each method is better in.

1.5 Project Plan and Contributions

This research project intends to apply the successful methods of RDR prudence to Internet banking fraud detection. Theoretically, RDR prudence seems to be precisely what fraud detection in Internet banking systems requires given the need to identify new fraud patterns instantly in this domain. This project will therefore investigate the performance (in terms of classification accuracy and prudence accuracy) of RDR prudence in online banking. Specifically, this will be achieved through the following stages:

1. Extend Single Classification RDM to Multiple Classifications RDM. The current RDM method (Prayote, 2007) uses a single class RDR engine. This project will redevelop the method to handle multiple classifications domains. Having a multiple classifications RDM system would enable comparisons between RDM and RM.

2. Evaluate and compare RM and RDM. There are currently no known direct comparisons of RM and RDM. This research will conduct focused comparisons of the two methods and give informed analysis of the particular strengths and weaknesses of each of the

methods. The evaluations will be done using simulated experts which will also be developed in this project.

3. Develop an integrated prudence method by combining RM and RDM. The combination of a structural and an attribute-based prudence method will provide some insight on whether the integrated method has better performance than the individual methods.

4. Evaluate and apply the best prudence method to Internet banking fraud detection. The practical and commercial contribution of this project will be in using a chosen prudence method on an Internet banking transaction dataset. The chosen method will be whichever has recorded best results from the evaluations mentioned in the second and third points.

1.6 Thesis overview

Chapter 2: Fraud Detection

Chapter 2 sets the scope of the research and provides a summarised review of the literature starting with Internet fraud, its effects and latest statistics and the industry most affected by this crime (financial sector). The chapter then analyses common trends in detecting frauds especially in Internet banking. Outlier Detection is then introduced as the fundamental aspect of all security mechanisms used to detect frauds in computer systems. Two applications of Outlier Detection namely Intrusion Detection and Fraud Detection are surveyed. The two are the basis of all commercial Fraud Detection and Intrusion Detection solutions. A profiling of three commercial Internet payment fraud detection systems reveals a common generic architecture. The majority of online banking fraud detection systems use a Rule-Base and an ANN. The profiled systems represent eight of the world's top 20 banks, are used in over 40 countries and in 430000 Internet banking systems. The chapter concludes with a cautionary note from an industry report warning that modern fraud detection systems will need to adopt intelligent programming techniques in order to improve their Knowledge-Based Systems (KBS) and algorithms for rapid detection of novel patterns.

Chapter 3: Knowledge-based Systems and Ripple Down Rules

Given that Rule-based systems are the main components of most fraud/intrusion detection systems, this chapter will give an overview of Knowledge-based or Expert Systems. The chapter also explains some of the main limitations of KBS including lack of causal knowledge,

slow knowledge acquisition process and lengthy and costly maintenance. RDR is then introduced as using a more efficient knowledge acquisition process, having an integrated rule addition and maintenance approach and essentially providing an alternative to the conventional KBS' limitations. A structured discussion of RDR is followed by the introduction of Prudence as a solution to KBS brittleness providing the ability to notify an expert or system administrator each time a new case or potentially wrong conclusion was produced. The chapter concludes with a review of two early attempts at prudence.

Chapter 4: Prudent RDR Methods

Building on the introduction and definition of Prudence in the previous chapter, chapter 4 analyses two of the newest and most successful prudence methods; RM and RDM. The chapter describes the structure, components and configurations of each method. RDM is originally a single classification method and this project extends the method to a multiple classifications version (MC-RDM). An explanation and the reasons for this extension are detailed in this chapter. The last part of chapter 4 introduces the Integrated Prudence Analysis (IPA) as a methodical combination of RM and RDM proposed by this reseacrh. IPA is a novel, experimental prudence method to improve RM and RDM's individual performances by combining an attribute based prudence method (MC-RDM) and a structural based prudence method (RM). Combining the two approaches has not been done before and is anticipated to take advantage of both the supplementary rule path context extraction of RM and partition based outlier detection of MC-RDM. As with RM and RDM in earlier sections, a detailed description of IPA's structure, components and configurations is given.

Chapter 5: Methodology

Chapter 5 explores different metrics and methods used to evaluate knowledge-based systems. The chapter reviews four KBS evaluation approaches focusing on different areas of the KBS. One method measures the accuracy of a KBS, another detects internal inconsistencies in the Knowledge Base and verifies if the system's actual behaviour is as specified in the design. Yet another method analyses differences between the KBS' false alarms and detection rates and the last method evaluates the Data Mining algorithms used in the KBS. Each of these methods is ultimately used to decide the degree of usefulness of a KBS by those who apply them. The chapter also presents the evaluation approach used in RDR based knowledge-based systems. The chapter describes the process of building a simulated expert and using the expert to evaluate a KBS under construction. A description of

the datasets and the metrics used specifically in this project is also given. The metrics, simple accuracy or classifier accuracy (Acc), Sensitivity (Se), Specificity (Sp) and prudence accuracy or Balanced Accuracy (BA) are used to determine a prudent system's precision in predicting correct domain cases and effectiveness of the system's warning mechanism.

Chapter 6: Results on Public Datasets

This chapter presents results and comparisons between RM and RDM on categorical and numerical public datasets based on the metrics discussed in chapter 5. The chapter analyses RM and RDM's classifier and prudence accuracy on the numerical and categorical datasets. Since the comparisons are meant to be a basis for selecting the better of the two methods, the chapter discusses a criterion for such a decision. The chapter also presents results from three Integrated Prudence Analysis (IPA) versions and evaluates the best IPA configuration against the two methods it was built from (RM and IPA). These evaluations between IPA, RM and RDM will determine if indeed a combined system has a better classifier and prudence accuracy than either of the constituent systems.

Chapter 7: IPA Results on Internet Banking Data

Results from IPA's evaluations on Internet banking transactions are presented in this chapter. A description of the depersonalised, online banking transactions are given including descriptions of the data's attributes. The chapter also presents statistics from a commercial online banking fraud detection system and for use as a benchmark in evaluating the IPA system. Chapter 7 concludes with a discussion on some of the deterrents to effective research-based fraud detection solutions and some generic recommendations to improving Internet banking fraud detection including a possible adaptation of the Integrated Prudence Analysis fraud detection system.

Chapter 8: Conclusion

Chapter 8 is a summary of the whole project and starts with a quantitative overview of the latest losses from Internet fraud and what industry is affected the most. An outline of the current approaches to curbing Internet banking fraud is also given, including the limitations of these approaches given the dynamic and sophisticated nature of Internet banking fraud today. The RDR technology is listed as a possible solution to these limitations, although hitherto unused in Internet banking. Towards the end, the chapter specifies a list of project

contributions and concludes with a note on the results, lessons learnt and future work in this domain.

2. Literature Review: Fraud Detection

2.1 Introduction

This Chapter introduces the foundational elements of this research. The chapter begins with an introductory overview of Internet fraud, elements of Internet fraud, statistical comparisons and trends of some of the persistent threats over the last five years. The financial sector, especially online banking is revealed as one of the industries most affected by Internet scams. To better understand Internet banking, a brief analysis of online banking is given, including main components of a generic online banking system and the general approach to fraud detection and security in these systems. Outlier detection is then introduced as the fundamental aspect of all security mechanisms used to detect fraud in computer systems. Two applications of outlier detection namely; intrusion detection and fraud detection are surveyed. The two are the basis of the majority of commercial fraud detection and intrusion detection software. The chapter concludes with a brief, informative overview of three commercial fraud detection systems and some insight from other researchers on merging fraud detection and intrusion detection ideas into a single, robust fraud and intrusion screening solution. Each concept in this chapter has been carefully explained to a level of detail appropriate for understanding the rest of the material covered in latter chapters. This was done to set a relevant depth of comprehension of all concepts and ideas discussed henceforth in the dissertation.

2.2 Internet Fraud

Internet fraud, online fraud and cybercrime are broad terms used to group various illegal activities committed through the Internet. A 2011 report commissioned by the British Cabinet Office defines cybercrime as illegal activities exploiting the Internet to illicitly access or attack information and services (Detica, 2011). The Australian Institute of Criminology (AIC) describes Internet fraud as any dishonest activity using the Internet as a target or means of obtaining some financial reward (Smith, 2001). Finally, online fraud is described by the Australian Federal Police as any type of fraud scheme using the Internet to conduct fraudulent transactions or transmit the proceeds of fraud (Australian Federal Police, 2012). The terms Internet fraud, online fraud and cybercrime are therefore often used interchangeably to describe various crimes, frauds and scams committed online. The list of

online fraud examples is ever growing and comprises phishing, hacking, malware dispensation, denial of service attacks, spoofing, identity theft, spamming and many more. One or more of these scams are used to achieve different fraudulent motives including financial gain.

The AFP website defines Internet banking fraud as theft committed using online technology to illegally remove money from a bank account, or transfer it to a different bank account (Australian Federal Police, 2012). One of the most prominent examples of Internet fraud is Phishing (RSA, 2010). Phishing involves the use of technology and deceit to illicitly acquire an unsuspecting victim's credentials (APWG, 2011). The sought after credentials include personal details, online banking usernames and passwords and other confidential credentials and details. With these credentials, fraudsters can easily gain access to their victim's email, social security benefits and bank account.

Phishing usually targets Internet users' online banking details and was responsible for losses of up to US$3.2 billion in the US alone in 2007 (McCall, 2012). In 2008, phishing was said to have cost banks more than US$3 billion globally (McCombie, 2008). A year later in 2009, the Australian Bureau of Statistics estimated that approximately 3.5% of the nation's population aged 15 years and over had been victims of some form of identity fraud or phishing (ABS, 2010). Although not all identity theft attempts and phishing campaigns are successful, the sheer volumes of the attempts alone indicate how relentless the fraudsters have become. Despite vigilant campaigns on safe Internet use by banks and other groups, a 2010 report by RSA, a renowned security company cautioned that fraudsters were equally at work, improving the technology and sophistication of their attacks (RSA, 2010). The report further forecasted that beyond 2010, there would be a rise in the effectiveness of phishing attacks resulting from new techniques.

The Anti-Phishing Working Group (APWG) received 315517 unique phishing emails reported by consumers in 2010 (APWG, 2010). In the same year the APWG detected up to 365628 unique phishing web sites. The most targeted industry in this period was the financial sector, receiving an average of 41.23% of all phishing attacks. The rest of the phishing targets included the retail industry, online classifieds, social networking, online auctions and the gaming industry (APWG, 2010). The following year (2011) saw a small decline in the number of phishing emails with 284445 reported to APWG. However, the number of phishing sites rose to 427314, an increase of 17% from the previous year. The financial sector remained the phishing campaign's most targeted industry sector, receiving on average, 44.75% of all

phishing attacks (APWG, 2011) . The increasing number of phishing sites and the growing proportion of attacks on the financial sector affirm the primary motive of most phishing campaigns, financial gain.

A number of conclusions could be drawn from the given results but the certain fact however is that phishing does not seem to be slowing down. In fact a variety of new phishing techniques are being invented and unleashed on unsuspecting and ignorant web users every day. The RSA fraud reports revealed that more diverse phishing attacks were deployed in 2011 (RSA, 2012), and that in the first half of 2012, the number of phishing attacks had increased by 19% (RSA, 2012). The report also adds that throughout the year 2011, financial institutions were the target of at least half of all phishing attacks.

2.3 Online Banking, a Victim of Internet Fraud

Online banking or Internet banking includes all traditional banking transactions conducted over the Internet through the user's banking institution's website. Through online banking, banks have been able to provide banking facilities to their clients over the Internet. However, this capability has not been easy nor cheap to maintain (Datamonitor, 2009). It is therefore unsurprising that the first reported phishing attack in Australia was against a bank in March 2003 (McCombie, 2008). Earlier in 2001, a group of bank executives and IT managers had advised a quantitative survey that one of the biggest challenges to online banking would be security (Aladwani, 2001). A typical online banking system will have the following subsystems with different access levels (RBI, 2001):

- Information only subsystem
- Electronic Information Transfer subsystems
- Fully Transactional subsystems

Information only subsystems provide non-sensitive information such as interest rates, branch locations, membership application forms and other general information. No identification of users is required with Information only subsystems. The Electronic Information Transfer subsystems provide read only, customer specific information including account balances, transaction records and account statements. Customers need to be identified and authenticated to access this part of the online banking system. Fully Transactional subsystems interact with the customer to update customer accounts such as

when funds are transferred to a different account or when a payment is made. The Fully Transactional subsystem requires customer identification details such as account number, login ID and password to access the right account and for authenticating a customer (RBI, 2001).

The Electronic Information Transfer and Fully Transactional subsystems form an important part of Online banking systems that are also commonly known as Electronic Payment Systems (EPS). EPS enable the electronic transfer of monetary value between a payer and payee (Sadeghi & Schneider, 2003). EPS have also been described as key tools for electronic commerce over the internet (Putland & Hill, 1997). Commerce, in this context, involves the exchange of money between a payer and a payee through a financial institution (Asokan, Janson, Steiner, & Waidner, 1996). A number of EPS are in use commercially including PayPal, BPay, EFTPOS, SecurePay and many more. These systems are used by banks and other financial institutions to effect transactions between trading parties.

Different research papers have provided a range of classifications for EPS. Some of these categorisations include pre-paid and pay-later classification, the digital currency and credit-debit classification and the cash-like and debit order categorisation (Asokan, Janson, Steiner, & Waidner, 1996; Sadeghi & Schneider, 2003; Havinga, Gerard, & Smit, 1996; Abrazhevich, 2001). Although the underlying details are almost the same in these classifications, the naming conventions are not uniform. This is affirmed by (Sadeghi & Schneider, 2003) that the various classification schemes are based on a number of aspects including: whether the payments are processed online or offline; whether a payment is pre-paid or pay-later and whether the payment system involves the use of some hardware token or not. This research will not delve into detail about specific EPS or what differentiates one from the other.

A principle that is almost unanimous in the publications reviewed in the previous section is the security requirements for EPS. Some of the most cited requirements include usability, confidentiality and availability (Sadeghi & Schneider, 2003; Asokan, Janson, Steiner, & Waidner, 1996; Putland & Hill, 1997). Usability is meant to ensure that paying or accepting payment through an electronic payment system is not a complex task for either the payer or the payee. Confidentiality requires that the details of the transaction should be exclusively restricted to the parties involved in the transaction. Availability ensures that the system (EPS) is available whenever required by the users.

The items discussed earlier are only a subset of the longer list of security requirements for EPS. (Asokan, Janson, Steiner, & Waidner, 1996; Putland & Hill, 1997) advise that not all EPS satisfy all these requirements and that it is not always essential to do so. It is also warned that the requirements may not be able to detect the legitimated user from a thief using legitimated login details (Putland & Hill, 1997). A user with a stolen username and password will have access to a particular account and can conduct a transaction from the compromised account. Despite the security features of electronic payment systems, a smarter, consistent and invulnerable layer of security is still essential to allow exclusive access to legitimate users and to detect illegitimate users in case of a breach.

2.4 Detecting Fraud with Outlier Detection

A variety of techniques are employed to identify fraudulent transactions and sense new trends in credit card systems, marketing systems, telecommunications systems and many other systems. Most of these techniques apply a concept usually referred to as Outlier Detection (OD). One of the earliest formal definitions of an outlier is that it is an observation that differs so much from other observations that it seems to have been generated by a different mechanism (Hawkins, 1980). Other definitions of outlier add that an outlier is a value that is very far from the middle of the distribution (Mendenhall, Reinmuth, & Beaver, 1993), or a value whose occurrence frequency is very low and further located from the rest of the values (Pyle, 1999), or simply an inconsistent observation relative to the bulk of other observations (Barnett & Lewis, 1995).

Aggarwal & Yu (2001) note that many OD definitions and algorithms overly depend on proximity between values to detect outliers, and that this is not sustainable for high dimensional data as it is sparse. The sparseness of the data therefore means some cases will be further away from others but they are not anomalous in any way. (Aggarwal & Yu, 2001) adds that increasing dimensionality of data makes it increasingly difficult to detect outliers. Consequently, finding outliers in such data using the proximity oriented methods becomes complex. To circumvent this issue in high dimensional data, (Aggarwal & Yu, 2001) recommends transforming the data into lower dimensional projections. A lower dimensional projection in this context is one where the density of the data is sufficiently lower than the average density. Data density is the proportion of elements (cases) to their covering space in a defined cluster (Breunig, Kriegel, Ng, & Sander, 2000; Kriegel, 2012). Outlier Detection for this scenario would then involve observing density distributions of the data projections

where an outlier would be a point in the lower dimensional projection existing in a local region of very low density (Aggarwal & Yu, 2001). According to (Aggarwal & Yu, 2001), this perception of OD is relevant and viable as most commercial applications of OD involve high dimensional data. It is for this reason that (Ben-Gal, 2005) advises that although some definitions of outlier are generic enough to be relevant in many applications, the exact definition depends on the structure of the data and the assumptions made by the detection methods.

Outliers occur for a range of reasons; these include human error, system faults, data entry/conversion error, deliberate fraudulent behaviour, instrument errors and in some cases, naturally occurring deviations (Hodge & Austin, 2004; Last & Kandel, 2001). The occurrence of outliers in data may be accidental or a deliberate attempt to contaminate the data or defraud the system. In some cases, outliers are not accidents or fraud attempts but correct information, however strange or exceptional (Last & Kandel, 2001). The importance of identifying outliers differs for different domains. In cases where outliers are accidental and caused by machine or human error, the outliers have to be isolated to avoid contaminating the data and misspecification of the data model. In some cases, the outliers represent fraudulent activity and will need to be identified at the earliest opportunity to enable preventative measures. In cases where the outlier is a correct representation of the information, it may be important to identify the outlier to inform system custodians of a new trend, new pattern or new limit. Other reasons for identifying and isolating outliers include avoiding biased estimations and improving the data quality (Ben-Gal, 2005; Last & Kandel, 2001).

There are two main approaches to OD: Univariate methods and multivariate methods (Ben-Gal, 2005; Last & Kandel, 2001). Univariate outlier detection methods evaluate each variable in isolation and do not consider the correlations between variables. Multivariate OD methods take into account the interactions and dependencies between variables and use these associations to identify outlying observations (Last & Kandel, 2001).

A separate taxonomy of OD techniques involves categorisation into parametric and non-parametric methods (Ben-Gal, 2005). Parametric methods often either assume that the data distribution is known or use statistical parameters to build a model. Any observations that deviate from this model are flagged as outliers (Ben-Gal, 2005; Hodge & Austin, 2004). Parametric methods are suitable for data with a known distribution but are not applicable for high dimensional data sets whose distribution may be complex to compute. Non-

parametric methods are model free and do not depend on a pre-determined data distribution.

The two taxonomies briefly introduced above are not the only classifications of OD methods. Other categorizations are given by Barnett & Lewis (1995) and Prayote (2007).

There are numerous OD techniques applied in different domains which will be explored later. A seemingly popular view in OD research is that no single OD method or technique can be applicable for all situations (Hodge & Austin, 2004; Aggarwal & Yu, 2001; Ben-Gal, 2005). Some of the main considerations for selecting a particular OD technique recommended by (Hodge & Austin, 2004) include selecting a technique that suits the data at hand (data attribute types for instance), distribution of the data and whether the data is labelled or not. The application domain also matters in terms of what kinds of outliers are expected and how they are dealt with once identified.

OD techniques are applied in a range of domains such as fault diagnosis in pipelines and space instruments; credit card fraud detection; computer network intrusion detection; marketing forecasts and loan applications fraud detection (Aggarwal & Yu, 2001; Ben-Gal, 2005; Hodge & Austin, 2004). The next section explores the applications of outlier detection in intrusion detection and fraud detection.

2.5 Applying Outlier Detection in Intrusion Detection

One of the notable applications of OD is in Intrusion Detection (ID) (Ben-Gal, 2005). ID is the process of monitoring attempted access, file modification or entry to a network or computer system (Patel, Qassim, & Wills, 2010; Jones & Sielken, 2000). The advances in technology and increasing popularity of network dependent devices and systems have caused an emergence of a range of exploitations on networks and computers systems. The dependence of business systems, health systems, defence, banking and all other systems on networks necessitates implementation of thorough, precise network monitoring systems to guard against detrimental exploitations. Detecting these exploitations or intrusions involves monitoring all unauthorised activity and detecting the use of illegally obtained log in credentials in a computer system or network. Intrusion Detection systems are the applications used to automate the ID process (Patel, Qassim, & Wills, 2010).

ID systems are usually grouped into two main categories: Network based and Host based systems (Patel, Qassim, & Wills, 2010; Kabiri & Ghorbani, 2005). Host based ID systems exclusively monitor the host (or server) for intrusions (Kabiri & Ghorbani, 2005). These systems will not detect nor attempt to observe intrusions beyond the host computer. Host based ID systems monitor such activities as local data requests, network connection attempts, login activity and read/write attempts and usually reside on the host computer (Kazienko & Dorosz, 2004; Patel, Qassim, & Wills, 2010). Although their dedication to a single host could be beneficial, (Patel, Qassim, & Wills, 2010) advises that Host based ID systems' tight integration to the host's operating system could prove problematic after operating system upgrades. The Network based ID systems are distributed along the network and monitor intrusions to all systems connected to the network (Kabiri & Ghorbani, 2005; Patel, Qassim, & Wills, 2010). Network based ID systems may be employed to monitor network firewalls, routers and client machines (Kabiri & Ghorbani, 2005; Kazienko & Dorosz, 2004).

A mixture of Host based and network based ID systems can also be applied on the same network as suggested by Kazienko & Dorosz (2004). In this approach, a blend of Host based and Network based ID systems may be considered to form a separate class of ID systems called Network Node Intrusion Detection Systems (NNIDS) or Hybrid per host ID systems. The arrangement is such that a Network based ID system monitors the whole network and each NNID agent is deployed on every host to process only the network traffic directed to that host. The choice of which ID approach to implement will therefore be dictated by the security needs, the size of network, the type of network and how the ID system is deployed.

ID systems usually apply one of two detection methods: Signature Detection or Anomaly Detection (AD) (Patel, Qassim, & Wills, 2010; Kabiri & Ghorbani, 2005). AD defines a model of normal (or acceptable) behaviour and detects as abnormal anything that does not conform to this model. AD is designed to detect patterns that deviate from what has been defined as normal behaviour (Patel, Qassim, & Wills, 2010). An ID system that uses an AD method would first define the normal, acceptable region or range. Any observations classified beyond this region will be treated as anomalous. Defining a distinctive region is not a trivial task in real applications. Some of the main difficulties according to Gonzalez & Dasgupta, (2003) include the imprecision of the boundary between normal and anomalous behaviour. Drawing a definite boundary that separates the two is therefore harder. Furthermore, in many domains, behaviour is dynamic, the normal and abnormal behaviours are regularly changing; a current normal may not necessarily be a normal forever. This means that regular

redefinitions of either behaviour are required (Gonzalez & Dasgupta, 2003). The other AD complicating factor is that the notion and degree of anomaly differs for different domains and may not be uniform for a single domain. For example, in online banking, a single transaction of $10000 may be anomalous in one account but normal in another. For these reasons, most AD techniques are heavily influenced by the application domain, availability of training and testing data and types of anomalies to be detected (Gonzalez & Dasgupta, 2003). Another cited problem with AD methods is their susceptibility to high False Positive rates because of their innate inability to define anomalous behaviour (Patel, Qassim, & Wills, 2010). False Positives in this context include all normal cases (behaviour) incorrectly classified as anomalous or abnormal.

An example of an anomaly based ID system is FlowMatrix (AKMA, 2011). FlowMatrix is a network AD system and monitors the records from routers and other devices to identify anomalous security incidents. The system processes and learns records from these devices for a period of 14 days and builds a behavioural model of the network. This is the normal region definition phase in an AD system. Incoming records are then compared to the defined (normal) models and anomalous events that deviate from these models are identified. FlowMatrix performs a continuous, automatic analysis of the network and detected anomalies are classified according to the type of breach (Denial of Service, malicious scans, Alpha Flows, and others) and logged to the user.

The alternative to AD is Signature Detection, also known as Misuse Detection (MD) (Patel, Qassim, & Wills, 2010). MD involves categorising known anomalies and exploitation patterns and comparing observed behaviour to the pre-defined patterns. In MD, attacks are represented in signatures such that known exploitations will always be detected (Patel, Qassim, & Wills, 2010). Exploitations in this context could be a bit string (e.g. virus bit string) or may describe a set or sequence of actions. The detection system monitors incoming observations and carefully compares them against these known patterns such that even variations of the known patterns should be detected (Patel, Qassim, & Wills, 2010). The main disadvantage with MD is that it will only detect the defined signatures or those that fit the generalised function(s). Effectively, unknown attacks cannot be detected if a function has not been defined for them. As new attacks are discovered, (Patel, Qassim, & Wills, 2010) remarks that signature databases should be constantly updated and the ID system should be able to keep up with the growing collection of signatures and also be able to conduct matches in a timely manner.

SNORT is a signature based ID system and highly regarded in many security forums. It is an open source, lightweight ID system and is reportedly the most widely used ID technology worldwide (SNORT, 2011). The system performs traffic analysis and packet logging on networks and detects worms, vulnerability exploit attempts, port scans, and many other suspicious incidents on the network (SECTOOLS, 2011). SNORT has three main components: the packet decoder, the detection engine and the logging and alerting subsystem. The packet decoder sets pointers into the packet for access and analysis by the detection. The detection engine maintains detection rules and uses them to decide whether a packet is suspicious or not. The first rule that matches a decoded packet triggers the specified action. The logging and alerting system logs decoded packet to a directory and sends alerts to a text file or as popup messages (Ditcheva & Fowler, 2005).

MD and AD have their strengths and weaknesses and selecting one of them depends on the domain and security needs of the organisation. The main thing to note according to (Patel, Qassim, & Wills, 2010) is that AD tries to detect the complement of good behaviour, whereas MD tries to detect known bad behaviour. In many cases, anomalous behaviour is often much smaller than normal behaviour, so it may be easier to list all known attacks and use a MD approach. However, if there is a clear, steady distinction between normal and anomalous regions, then it may be wiser to define the normal region and implement an AD approach. To get the best of both MD and AD techniques, the two may be used together in a single system or in a complementary manner (Jones & Sielken, 2000).

The Haystack is one of the early ID systems that combines both AD and MD approaches. Haystack utilises users' log profiles to detect intrusions. The system employs both AD and MD and detects a number of intrusions including Denial of Service attacks, attempted break-ins and information leakage (Smaha, 1988). The MD component part of the system has a set of predefined suspicious behaviour patterns. The system also has a suspicion quotient used to determine the level of abnormality of a detected attack. If a pattern matches the predefined "bad" behaviour patterns and has a suspicion quotient higher than the defined threshold, then it is detected as an anomaly. The AD part of the system monitors 24 features of a user's session and uses past statistics to detect significant deviations from the recorded statistics (Smaha, 1988).

Intrusion Detection undoubtedly has immense value and importance in the security of networks and computer systems. As the networks grow and the technologies that support them advance, so do the exploits and their sophistication. A number of challenges facing ID

systems have been given. A technical report published in 2000 advised that newer ID systems had to be able to manage and process large volumes of data in acceptable times (Jones & Sielken, 2000). The report further advised that new ID systems should be able to detect as much anomalous behaviour as possible and have real time detection capabilities (Jones & Sielken, 2000). Five years later in 2005, a journal article by Kabiri & Ghorbani (2005) added that ID systems were not fully reliable yet and that the ability to detect novel patterns had to be immediately addressed. A more recent journal article by(Patel, Qassim, & Wills (2010) maintains that ID systems need to adapt intelligent programming techniques and knowledge based systems to improve detection rates, handle significant amounts of data and boost computational power to be able to detect intrusions in real time. The article further recommends that the human effort needed to build and run these systems should also be reduced.

2.6 Outlier Detection in Fraud Detection

Another significant application of Outlier Detection is in Fraud Detection (FD) (Ben-Gal, 2005). A fraud, in a computer systems context includes any deliberate abuse or misuse of a system (Kou Y. , Lu, Sirwongwattana, & Huang, 2004; Phua, Lee, Smith, & Gayler, 2005). Fraud Detection is the monitoring of users' interaction with the system to estimate and detect undesirable behaviour (Kou Y. , Lu, Sirwongwattana, & Huang, 2004) or in simpler terms, the automation of a fraud screening process (Phua, Lee, Smith, & Gayler, 2005). FD systems are used in different domains to detect different frauds including credit card fraud, online banking fraud, telecommunications fraud and online insurance fraud (Bolton & Hand, 2002; Kou Y. , Lu, Sirwongwattana, & Huang, 2004).

Fraud Detection systems is a generic term given to systems used to detect fraudulent behaviour. A number of approaches and technologies for FD systems have been suggested and reported in different research and technical publications. Different taxonomies for these technologies have also been suggested by researchers in this area. For example, Phua, Lee, Smith, & Gayler (2005) categorises approaches to FD in terms of supervised methods, semi-supervised methods, unsupervised methods and hybrid methods. The organization employed by Kou, Lu, Sirwongwattana, & Huang (2004) involves first categorising the different frauds namely credit card fraud, computer systems intrusion and telecommunications fraud and then detailing the techniques used to detect fraud in each of the categories. The techniques surveyed under each category include Neural Networks,

Expert Systems and Data mining (Kou Y. , Lu, Sirwongwattana, & Huang, 2004). In Bolton & Hand (2002)'s review paper, a range of statistical based FD techniques are analysed under five categories of fraud. These categories are money laundering, computer intrusion, medical and scientific fraud, telecommunications fraud, computer intrusion and credit card fraud (Bolton & Hand, 2002). The large variety of available FD systems and techniques enables different researchers to choose a taxonomy and categorisation of their choice when reviewing one or a group of these methodologies. Some of the commercial FD systems used in banking systems include the Falcon Fraud Manager, the ACI Proactive Risk Manager (PRM) and the SAS Fraud Manager. A brief review of each of these systems follows. It must be noted that the listed systems are proprietary and in commercial use. Consequently, the amount of public detailed and technical information about these systems is therefore limited.

The Falcon Fraud Manager is one of the most widely used and most accurate fraud detection systems by banks worldwide according to a report released by its developers (FICO, 2011). Falcon is mainly used for payment card fraud detection and has real time detection capabilities (FICO, 2011). Falcon uses an anomaly detection technique by profiling cardholders' transaction behaviours to spot uncharacteristic behaviour. A diagram of the Falcon's architecture shows that the system has a Rule Base and a Neural Network as some of its main components (FICO, 2011). Figure 2-1 shows an abstracted version of the Falcon system showing the interaction between the neural network and the Rule-base.

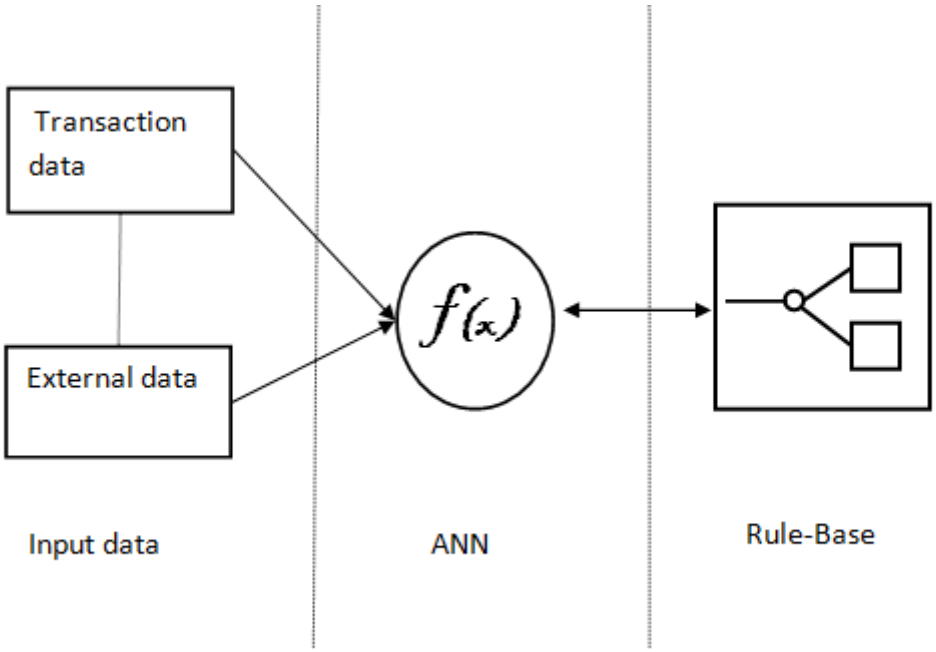

Figure 2-1. Some of the Falcon s main components (FICO, 2011)

Like Falcon, the PRM is reported to be commercially successful and is used in over 40 countries including eight of the top 20 banks in the world (ACI Worldwide, 2011). PRM is a debit card, internet banking and credit card FD solution and is also reported to have real time transaction monitoring, helping to prevent loss of funds (ACI Worldwide, 2011). PRM employs both anomaly detection and signature detection methods; each transaction is compared with the custom fraud model and also with recorded behaviour patterns of account holders. In terms of architecture, neither of the two PRM white papers provide detail on the specific structure of the system, except that it combines an expert Rule Base with a Neural network (ACI Worldwide, 2011; IBM, 2008).

Reportedly used at 43000 sites, the SAS Fraud Management system is another commercial FD solution (SAS, 2007). A debit and credit card FD solution, the SAS Fraud management system also boasts real time detection capabilities. The SAS system uses an anomaly detection method. Users' transaction patterns are profiled into unique 'signatures' and unexpected deviations from these signatures are reported to the system administrator (SAS, 2007). The system's architecture includes a Rule Base and an ensemble of Self Organising Neural Networks (SONNA) (SAS, 2007).

The consistent use of Rule Bases and Neural Networks in all of these FD systems is unsurprising. One of the earliest FD systems used at the Mellon Bank in New York used a Neural Network (Ghosh & Reilly, 1994). Cardwatch, another of the early FD systems also

used a Neural Network to process current transaction patterns and detect possible frauds (Aleskerov, Freisleben, & Rao, 1997). A steady advocacy and report of widespread use of Rule Bases and Neural networks is also given by (Bolton & Hand, 2002; Kou Y. , Lu, Sirongwatana, & Huang, 2004; Phua, Lee, Smith, & Gayler, 2005; Weatherford, 2002). The next Chapter will take a closer look at Rule Bases and Neural Networks and analyse their strengths and weaknesses and when it is ideal to use each.

Other approaches suggested for use in Internet banking fraud detection and reported in academic forums include the use of a Hidden Markov Model (HMM), a hybrid model comprising One Class Classification (OCC) and a Rule-Base and ContrastMiner, another hybrid system consisting of neural networks, a contrast pattern mining module and a decision forest (wei, Li, & Cao, 2012; Mhamane & Lobo, 2012; Krivko, 2010). Although the publication on using HMM cites the success of a similar approach in a separate study, the publication provides no results and is more of a proposal and justification for using HMM (Mhamane & Lobo, 2012). Krivko (2010)'s hybrid approach is meant to bridge the individual limitations of supervised and unsupervised methods. The approach has two fraud detections levels; the OCC component sits on the first level and monitors changes in user behaviour and assigns a score based on the level of deviation to normal behaviour. In the second level is the rule-base filter which processes cases that have been passed from the first level. Cases that violate any of the rules are marked as suspicious. Wei et al (2012)'s approach also features a hybrid system including a contrast pattern miner, a neural network and a decision forest. Typical of hybrid systems, the system's strength lies in leveraging a host of data mining models and is reportedly impressive in the skewed Internet banking data (Wei, Li, & Cao, 2012). The two hybrid systems Krivko (2010) and ContrastMiner were able to detect around 57% and 66% respectively of all frauds.

Traditionally, Intrusion Detection and Fraud Detection have been separate fields, addressing separate security needs. Most ID research has originally focused on preventing illicit use at network level and Operating System level (Patel, Qassim, & Wills, 2010; Kabiri & Ghorbani, 2005; Jones & Sielken, 2000). Conversely, FD research has also focused on detecting frauds at an application level (Bolton & Hand, 2002), (Kou, Lu, Sirwongwattana, & Huang, 2004). For example, some of the particular FD applications reviewed earlier sought to detect fraudulent behaviour in credit card systems, telecommunications systems and other specific applications (Kou, Lu, Sirwongwattana, & Huang, 2004; Phua, Lee, Smith, & Gayler, 2005). It is therefore fair to posit that there has not been much crossover between these two fields

although they rely on and apply the same principles. This is a view also held by (Kvarnstrom, Lundin, & Jonsson, 2000), who also suggest that merging FD and ID techniques may be a viable solution to protecting both networks and applications.

Other perspectives on Intrusion Detection and Fraud Detection are that most forms of intrusions are actually instances of fraud (Fawcett & Provost, 1997). In addition, Fawcett & Provost (1997) add their proposed FD framework can also be justifiably applied to ID. The Java Agents for Meta-Learning (JAM) project is a valuable substantiation of how FD techniques can be used and still be as efficient in ID applications (Stolfo S. , Fan, Lee, Prodromidi, & Chan, 2000). A possible hindrance to the escalation of FD-ID combinatory systems such as JAM is that the commercial market has established brands in each of these two fields so crossing into unfamiliar territory may not be justifiable in business terms. However, it is hoped that research will continue to inform on the methods, strategies and benefits of incorporating ID in FD systems or *vice versa*.

A host of varying mechanisms have been suggested to prevent fraud in online banking. One of the pivotal aspects of security in online banking according to Nilsson, Adams & Herd (2005) is trust. Nilsson, Adams & Herd (2005) assert that users' perceived trust of an online banking system is of vital importance given the persistent threats of phishing and other risks. A recommendation by Nilsson, Adams & Herd (2005) is that a balance should be struck between the user's trust of an online banking system, the actual security of the system and its usability.

An alternative perspective is that online banking websites should invest in browser security (Sood & Enbody, 2011). According to Sood & Enbody (2011), declarative security in Hyper Text Transfer Protocol (HTTP) response headers would be an additional defence mechanism and has been shown to reduce security flaws in online banking website. A 2007 paper by Mannan & Oorschot (2007) argued that the real issue in online banking is security and usability. It was reported that the security requirements for safe online banking were too difficult and unrealistic for average users and that there was a substantial difference between the banks' expectation of their online customers and what the customers were actually doing in regards to online banking precautions (Mannan & Oorschot, 2007).

2.7 Remarks and Observations

The view of this thesis is that sufficient and specific knowledge on commercial banks' security and fraud detection mechanisms is scarce. Moreover, banks rarely publish performance statistics of their fraud detection systems. Consequently, only bits and pieces of information on the banks' online banking infrastructure make it into the public domain. This proprietary information is usually kept confidential for commercial and security purposes. Banks are businesses foremost, and have to protect their commercial interests from competitors.

Secondly, most security software is provided by third party software companies who also want to protect their intellectual property from competing software vendors. Finally, both banks and security software vendors do not publicise most of the information on security systems to prevent attacks from technically savvy fraudsters. This secrecy and unavailability of information (despite the commercial reasons) slows the progress of research in this field. The same sentiment was also expressed by Bolton & Hand (2002) who assert that the exchange of ideas in fraud detection is limited and this hinders the development of new methods. Bolton & Hand (2002) also agree that it may not be wise to publicise detailed information on fraud detection techniques as this would give the fraudsters the information they need to elude detection mechanisms.

2.8 Chapter Summary

This chapter introduced the concepts on which the rest of the ideas and propositions of this thesis are based. To address the concern of fraud, especially phishing in Internet banking, two prominent applications of outlier detection were analysed. These two models; intrusion detection and fraud detection are at the core of many research ideas and indeed all commercial solutions to all sorts of fraud detection in computer networks and different applications. There appears to be a common approach to fraud detection in the Internet banking industry as evidenced by the three profiled fraud detection systems. All these systems are quite similar; they mainly feature a neural network and a Rule-base. This approach is consistent across all the reviewed systems, except for the unknown details rarely exposed by the developers nor end-users of these systems i.e. banks or the makers of the software themselves. The reason for this secrecy was noted and expressed in the previous

section and is corroborated by some research publications in this field such as Bolton & Hand (2002). Since Rule-bases are the major component of these systems and many other fraud detection applications, the next chapter takes a deeper look at what Rule-bases are. The chapter surveys what constitutes Rule-bases, what larger classification or category they comprise, any known issues they have and how these issues have been addressed.

3. Literature Review: Knowledge Based Systems and Ripple Down Rules

3.1 Introduction

The previous chapter presented an overview of the current approaches in detecting frauds in different domains. The chapter also revealed some of the challenges facing modern fraud detection systems in general. Some of the desirable innovations in fraud detection include the use of intelligent programming techniques, Knowledge-Based Systems (KBS) and accurate algorithms for rapid detection of novel patterns. It was also shown in the previous chapter that most commercial fraud detection and intrusion detection systems are based on a KBS and ANNs. In this chapter, KBSs are more closely examined including their generic architecture and their limitations. The chapter also provides an analysis of the alternatives to these limitations.

3.2 Knowledge-based Systems

The Macquarie Australian National Dictionary defines knowledge as facts, truth or principles usually from study or investigation (Macquarie Australia's National Dictionary, 2001). The same dictionary defines expertise as special skill or knowledge in a particular field. In a similar sense, Giarratano & Riley (2005) add that expertise is rare and specialised whereas knowledge is more general and low level. Following these assertions, Knowledge Based Systems (KBS) are generic systems that attempt to understand and imitate human knowledge (Wiig, 1994) while Expert Systems (ES) are specialised KBS that use knowledge to solve problems that are difficult enough to require human expertise (Giarratano & Riley, 2005). An alternate definition of ES is that they are computer systems/programs designed to model the problem solving ability of a human expert (Durkin, 1994) or systems that emulate the decision making ability of a human expert (Giarratano & Riley, 2005). Despite the small disparities, KBS, ES and Knowledge Based Expert Systems (KBES) are often used synonymously (Prayote, 2007; Giarratano & Riley, 2005) in reference to either system.

An ES mainly comprises a Knowledge Base (KB) and an Inference Engine (IE) (Liao, 2003; Li, Xie, & Xu, 2011). All data, rules, cases and relationships are stored in the KB (Abraham, 2005). The KB houses all the reusable knowledge used by the IE to produce conclusions (Li,

Xie, & Xu, 2011; Giarratano & Riley, 2005). During its use, the ES accepts some kind of query from the user through an interface. The IE uses the query parameters and the contents of the KB to generate a response which is then relayed to the user. The IE specifies the process through which output in the form of facts, conclusions or classifications is extracted from the KB (Hayes-Roth & Jacobstein, 1994). Figure 3-1 shows a schematic of a KBS.

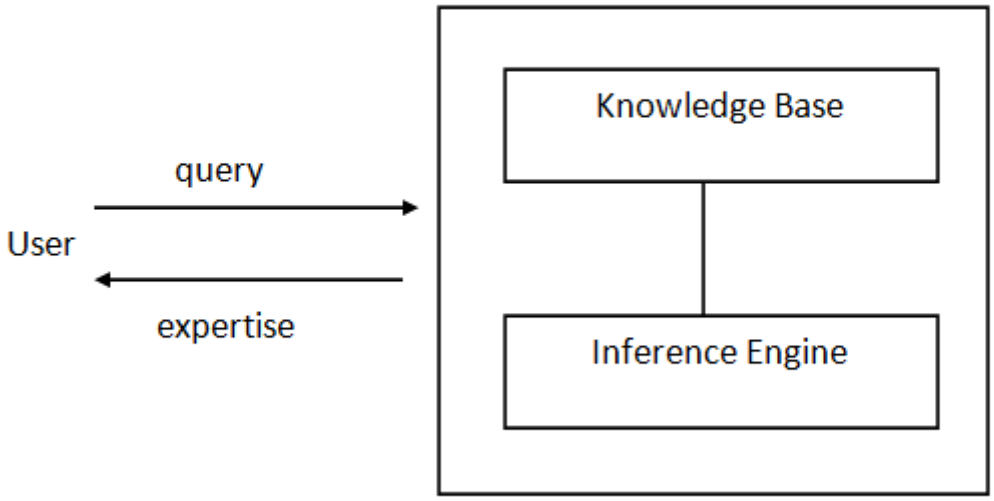

Figure 3-1. Basic schematic of a KBS (Giarratano & Riley, 2005)

KBS have been in use for more than 30 years since the development of the first one around 1980 (Hayes-Roth & Jacobstein, 1994; Feigenbaum & Buchanan, 1978). From their earliest inception, KBS were reported to have reduced goods and services production times dramatically and increased productivity.

Some of the examples of early KBS impressive business value include the SXEL expert system, known to have reduced a three hour task to 15 minutes (Hayes-Roth & Jacobstein, 1994). This productivity boost is said to have been annually worth $70 million in monetary terms. In another example, an early American Express credit screening system is said to have reduced the rate of incorrect credit refusals by one third, resulting in an additional $27 million per annum (Hayes-Roth & Jacobstein, 1994). Given that both of these facts were reported in 1994, the technology of KBS has since advanced tremendously and the power and speed of computers has soared significantly. Some other recently published benefits of KBS are that they capture and preserve human experience, they are more consistent than humans, they

can compensate for the loss of human experts and are much faster than human experts (Li, Xie, & Xu, 2011; Abraham, 2005).

The use of KBS has spanned various domains and industries including finance, military, construction, media, retail, government departments and many other areas (Hayes-Roth & Jacobstein, 1994). KBS are said to be especially effective in applications where knowledge rather than an imperative algorithm is needed to derive the solution (Preece, 2001). Some commonly listed early KBS include DENDRAL (Feigenbaum & Buchanan, 1978) and MYCIN (Buchanan & Shortliffe, 1984; Lindsay, Buchanan, Feigenbaum, & Lederberg, 1993). DENDRAL was used in elucidating chemistry to assist chemists in searching for molecular structures (Feigenbaum & Buchanan, 1978; Feigenbaum & Buchanan, 1993). MYCIN was built to provide advice on bacterial infections of the blood and could explain its reasoning (Buchanan & Shortliffe, 1984; Hayes-Roth & Jacobstein, 1994).

One of the newer applications of KBS is in online banking fraud detection. In 1997, a number of research publications were predicting that retail banking transactions will be available to customers online (Barwise, 1997; Booz, Allen, & Hamilton, 1997). By the end of 1997, one of the four major banks in Australia had Internet banking services (Sathye, 1999). Just over 10 years later, all the major banks in Australia provide online banking services with KBS powered fraud detection systems. Some of the online banking fraud detection systems in use today include Falcon, PRM and SAS Fraud Manager as profiled in Chapter 2. Each of these systems uses a Rule-Base System, one of the most popular types of KBS in use today (Giarratano & Riley, 2005).

3.3 Rule-Based Systems

Rule-Based Systems (RBS) have been regarded as an effective means of codifying human expertise (Hayes-Roth, 1985). RBS are a type of KBS where knowledge is presented in terms of multiple rules specifying what action or conclusion should or should not be taken in different situations (Giarratano & Riley, 2005). Hayes-Roth observed that experts tended to express their problem-solving techniques in terms of situation-action rules and that consequently, RBS should be the *de facto* KBS method (Hayes-Roth, 1985). In RBS, human expertise is presented as conditional rules that are used to provide a conclusion/action/answer to a contextual problem. Each conditional rule links the given condition(s) to a conclusion/action/answer (Abraham, 2005). Hayes-Roth adds that the

knowledge contained in RBS includes conditions for achieving some goal, specific inferences resulting from specific observations, probable causes of a set of symptoms and likely consequences of hypothetical situations (Hayes-Roth, 1985).

Since RBS are a type of KBS, the main components of a RBS are a Knowledge-Base (KB) and an Inference Engine. These components typically perform the same functions as in a generic KBS. The KB, also known as the rule-base in this context, is a collection of if-then rules and encodes the conclusions/actions/answers the system should produce given a specific situation. The IE applies appropriate rules to the current situation (Giarratano & Riley, 2005). A rule mainly comprises an antecedent (the situation) and a consequent (the conclusion/action/answer). If a given antecedent (the If part) is satisfied, the appropriate consequent (the Then part) is invoked (Hayes-Roth, 1985). Figure 3-2 shows the general structure of a rule.

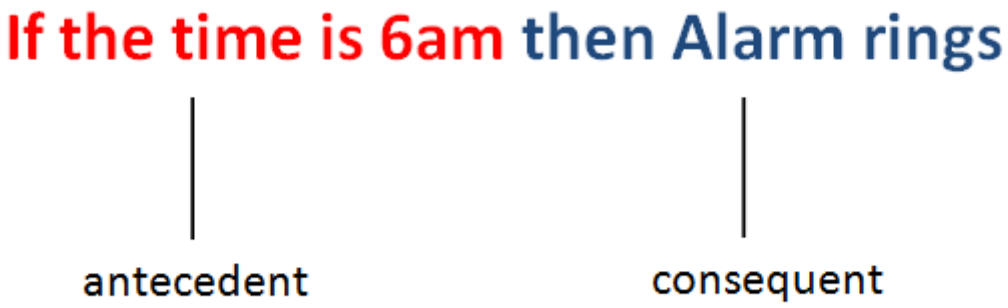

Figure 3-2. Components of a simple rule

Given the consistent use of KBS in different outlier detection applications, it follows that most fraud/intrusion detection systems in use today include some example of a knowledge base in their architecture. Since their introduction more than four decades ago, the advantages of KBS have been undoubtedly significant to say the least. Some of the main benefits stated earlier in this chapter include their ability to preserve human knowledge and experience, their superior consistency to humans and their running speed (Li, Xie, & Xu, 2011; Abraham, 2005).

In spite of their immense importance in different domains, KBS also have some considerable shortcomings. One of the main criticisms of KBS is their lack of causal knowledge. Without the understanding of underlying cause and effect, KBS cannot extend available information

to deduce new facts (Giarratano & Riley, 2005; Elkan & Greiner, 1993). For example, without explicit specification, an ES will not know that age cannot be negative and that only females can actually be pregnant. Another limitation of KBS is based on the knowledge acquisition bottleneck phenomenon, where the process of transferring knowledge to an ES is indirect (involves an expert and knowledge engineer), labour intensive and usually restricted to a specific context (Compton & Jansen, 1988; Giarratano & Riley, 2005; Dazeley, 2007; Richards, 2009).

Ignoring the dynamic nature of knowledge by traditional KBS wrongly assumed that ultimately, the domain specialist and knowledge engineer would produce a perfect image of the expert's knowledge. This approach to building knowledge-based systems is often criticised as being restrictive and costly (Hayes-Roth & Jacobstein, 1994; Dazeley, 2007). In some cases, the traditional approach to knowledge acquisition has even been labelled less innovative (Richards, 2009).

Under the traditional approach, changes of any kind to the knowledge base would always involve the domain expert and the knowledge engineer. Furthermore, such changes will most likely require the original expert to ensure that the latest changes do not render the prior knowledge invalid. These factors, and the fact that in these KBS, maintenance is performed as an additional task to knowledge acquisition (Richards, 2003), ultimately results in maintaining such systems being a time consuming and costly exercise (Hayes-Roth & Jacobstein, 1994; Dazeley, 2007). Figure 3-3 shows the change in complexity of a single rule in a medical expert system over three years (Compton & Horn, 1989). The rule changes from comprising just four conditions to being a compound of 28 conditions. This represents a threat to the knowledge engineer's understanding and ability to edit such a rule (Dazeley, 2007) and the general growth and continuity of the KBS.

1984

```
RULE(22310.01)
IF (bhthy or utsh_bhft4 or
vhthy)
    and not on_t4
    and not surgery
    and (antithroid or
hyperthyroid)
THEN
DIAGNOSIS("…thyrotoxicosis")
```

1987

```
RULE(22310.01)
IF ((((T3 is missing)
  or(T3 is low and T3_BORD is low)
  and TSH is missing
  and vhthy
  and not (query_t4 or on_t4 or
surgery or tumour
  or antithyroid or hypothyroid or
hyperthyroid))
  or((((utsh_bhft4 or
  (hithy and T3 is missing and TSH is
missing))
  and (antithyroid or hyperthyroid))
  or utsh_vhft4
  or ((hithy or borthy)
  and T3 is missing
  and (TSH is undetect or TSH is
low)))
  and
  not on_t4 and not (tumour or
surgery)))
  and (TT4 isnt low or T4U isn't low)
THEN DIAGNOSIS("…thyrotoxicosis")
```

Figure 3-3. Change in complexity of a rule over 3 years. The rule started with four conditions (left) and eventually had 28 conditions (right) (Compton & Horn, 1989)

3.4 Ripple Down Rules

Ripple Down Rules (RDR) was introduced around 1988 (Compton & Jansen, 1988; Kang, Compton, & Preston, 1995) as an alternative to the traditional KBS. RDR eliminates the need for a knowledge engineer as the expert directly interacts with the system (Kang, Compton, & Preston, 1995). Additionally, maintenance in RDR has been described as trivial and brief (Kang, Compton, & Preston, 1995). In RDR, maintenance and knowledge acquisition are essentially integrated and usually do not require the additional services of a knowledge engineer (Richards, 2009). In fact, one of the earliest commercial applications of RDR, a system named Pathology Interpretative Expert Reporting System (PIERS), was described as user maintained and not requiring knowledge engineering expertise (Edwards, Compton, Malor, Srinivasan, & Lazarus, 1993). The PIERS system was used to generate clinical

conclusions for pathology reports. In 2009, PIERS was reported to have processed about 30 million reports and been used by 14 pathology laboratories (Richards, 2009). Other applications of RDR extend to Web browsers, help desk systems, online shopping, email management systems (Kang, Compton, & Preston, 1995; Richards, 2003) to cite a few. Some of the latest applications of RDR includes in an eHealth document management system, in Agile Software development for Expert Systems, in a home-based Telehealth system and most recently in internet banking fraud detection, which is one of this research's contributions (Maruatona, Vamplew, & Dazeley, 2012; Dazeley, Park, & Kang, 2011; Yoon, Han, Kang, & Park, 2012; Han, et al., 2013).

RDR has a binary tree structure in which each node corresponds to a rule. The root node, which is always true by default, is connected to a network of nodes, also connected to their parent nodes through either a false or true branch. Every parent node has two possible branches; the true and false branches. The parent is connected to a child node through the branch that represents the evaluation of the parent's rule. For example, if the parent node's rule evaluates to true for a given case then the child connected to the parent through the true branch is evaluated next. The same case applies for a child node on the false branch if the rule evaluation returns false. Ultimately, the terminating node is reached after all its parents' nodes have been evaluated. As newer, more specific rules are added to the lower levels of the tree, the broader rules get evaluated first, leading to the newer rules. This rippling from generic rules to specific ones is what earned the method its Ripple–Down name. The rippling continues until a terminating/leaf node is reached. The conclusion from the last successful node is returned as the ultimate conclusion for the case. If the last firing rule was false, then the last true ancestor's conclusion is returned as the effective conclusion. RDR's root node has a default conclusion such that if no other rules evaluate to true, the default conclusion is returned. This guarantees that a conclusion will be returned every time. Figure 3-4 shows an RDR tree with a case A= {a, c, r, t}.

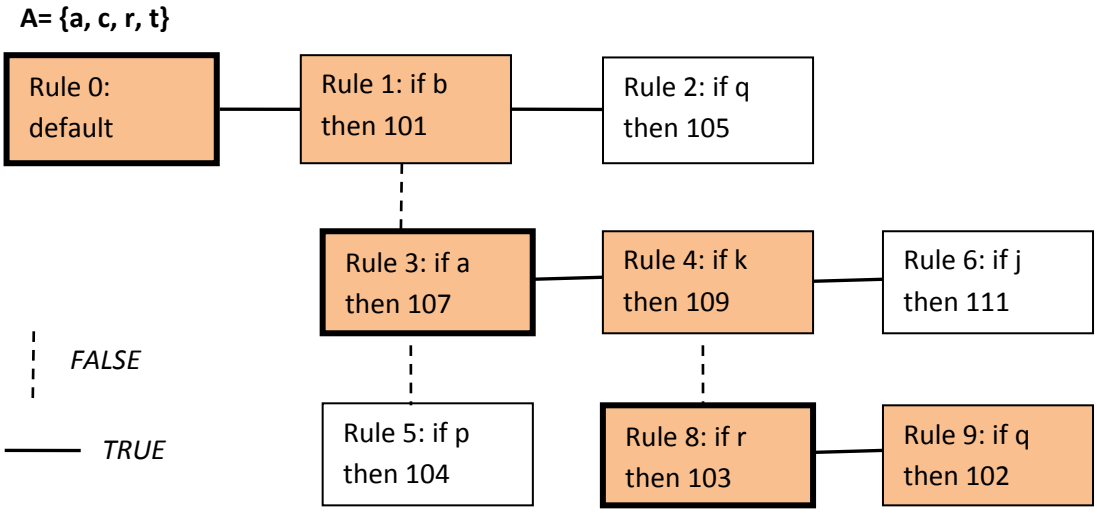

Figure 3-4. The RDR tree structure. Horizontal solid lines represent true branches and false branches are represented by vertical dashed lines. Input A is the current case being processed. The shaded boxes represent evaluated nodes. Nodes which evaluated to true are represented by bolded boxes

The bolded rectangles in the RDR tree represent rules that return true for the case A= {a, c, r, t}. Rule 0 is the default rule so will always return true. For this case, inferencing follows the highlighted nodes. The first rule tested after the default node is Rule 1, this node evaluates to false since the condition does not satisfy any of the case attributes. The false branch of Rule 1 leads to Rule 3. Rule 3 evaluates to true for the current case so Rule 4 is evaluated next. Rule 4 evaluates to false and leads to Rule 8, which returns true and leads to Rule 9. Rule 9 is false so the effective conclusion is 103 from Rule 8 since this is the last tested rule that returned true.

3.5 RDR Learning

One of RDR's advantages over other KBS methodologies is its contextual oriented rule addition, whereby any new rule is added and validated within a specific context (Dazeley, 2007). To accommodate the dynamism of knowledge, Richards (2009) advises that knowledge acquisition must be incremental and captured within the context in which it is valid. This gives RDR the flexibility to make specific amendments to the Rule-base without invalidating the previously captured knowledge. The goal of RDR, according to Kang, Compton, & Preston (1995), is to incrementally build a knowledge base where new rules are refinements to older rules added in a way that does not nullify pre-existing knowledge. RDR

learns as new rules are added to the knowledge-base. A new rule is added if the expert disagrees with the system's conclusion. The expert then provides a justification of why the system's conclusion is wrong, and the justification is the basis of the new rule (Compton & Jansen, 1988). Each new rule is added to deal with a specific case so the expert's justification is formed by comparing the new case with the case that caused the creation of the last firing rule for the present case. The former case is known as the cornerstone case and each rule has a directly associated cornerstone case. The case is being processed at any moment is known as the current case. To guarantee that the new rule is exclusively satisfied by the current case, the expert selects one or more attributes from a difference list (Kang, Compton, & Preston, 1995). A difference list consists of attributes from the cornerstone case and the current case and ensures that the new rule satisfies the current case and not the cornerstone case. For example, consider two models of the BMW 1 Series models. The two models' attributes are given below (Car Reviews, 2012):

Model	16I Sport	18I Sport
Engine Type	Turbo	Turbo
Engine Size	1.6 L	1.6 L
Max Torque	220 Nm@ 1350rpm	250 Nm@ 1500rpm
Max Power	100kW @ 4400 rpm	125kW @ 4800 rpm
Fuel Type	Unleaded Petrol	Unleaded Petrol
Valve Gear	Variable Overhead	Variable Overhead

Table 3-1. Attributes of two models of sports cars, the 16I and the 18I

Assuming some RDR based car comparison system wrongly classifies the 18I Sport as the 16I Sport, a typical difference list for the current case (18I Sport) and the cornerstone case (16I Sport) may be formulated as given below.

Case	Cornerstone	Current
Max Power	100kW @ 4400 rpm	125kW @ 4800 rpm
Max Torque	220 Nm@ 1350rpm	250 Nm@ 1500rpm

Table 3-2. A sample difference list of two different editions of sports cars, the 16I (cornerstone case) and the 18I (current case)

The expert can select the conditions for the new rule as Max Power == 125kW @ 4800rpm and Max Torque == 250Nm @ 1500rpm. This rule will be guaranteed to work on the new case (18I Sport) and not for the 16I Sport since the 16I Sport does not have neither a Max

Torque of 250Nm @ 1500rpm nor a as Max Power of 125kW @ 4800rpm. The new rule will have the 18I Sport's attributes as its cornerstone and will be added as a child node of the 16I Sport. If a new rule describing a new diesel powered version of the 18I were to be added, then a similar process would be followed and so forth. The context specificity of the rules also means that the rules are validated at the time they are added, effectively eliminating a need for an additional validation task (Dazeley, 2007).

Despite its advantages over traditional KBSs in terms of its knowledge acquisition and maintenance methodology, RDR's usability has also been questioned, specifically on three main aspects. It has been argued that the contextual nature of rule addition in RDR can result in repetitions of the same knowledge in different parts of the tree (Compton & Horn, 1989). However, it has since been found that this is not a major problem after tests showed that less than 15% of the knowledge was repeated when an expert system was built using RDR (Dazeley, 2007; Kang, Compton, & Preston, 1995). The other misgiving of researchers about RDR is in relation to the possibility of the tree structure being unbalanced due to the lack of structural control of the knowledge base by the expert or knowledge engineer trees (Compton, Kang, Preston, & Mulholland, 1993). Again, this criticism has been found to have no real impact after an RDR tree was found to have only twice as many rules as other optimised induction (Kang, Compton, & Preston, 1995).

The last aspect of RDR's shortcoming is its inability to provide multiple classifications. A practical demonstration of this limitation was shown by PIERS' (Edwards, Compton, Malor, Srinivasan, & Lazarus, 1993) inability to provide multiple diagnoses in cases where patients have multiple diseases (Kang, Compton, & Preston, 1995). A possible workaround to this inefficiency would be to separate the problem into sub-domains and have an independent KB for each sub-domain. However, it is difficult to separate sub-domains in practice so this would be a difficult task for many domains (Kang, Compton, & Preston, 1995). So far, the accepted alternative solution to this problem is an extension of the RDR structure in a way that allows it to generate multiple classifications. This idea is explored further in the next section.

3.6 Multiple Classifications RDR

Multiple Classifications Ripple Down Rules (MCRDR) is an extension of RDR with the added capability to handle multiple classifications (Kang, Compton, & Preston, 1995). The inability

of RDR to provide multiple classifications was seen as a limitation to its applicability in many domains, a classic example being the failure of PIERS to provide more than one diagnosis (Kang, Compton, & Preston, 1995). MCRDR was therefore developed to be applicable for domains where a single case may lead to multiple conclusions while still retaining the advantages of RDR. In fact, MCRDR has been shown to cover a domain quicker than its single class counterpart and has also been reported to produce a more compact KB with fewer redundancies than single class RDR (Richards, 2009).

One of the main differences between single classification and multiple classifications RDR is that MCRDR's structure is an n-ary tree compared to RDR's binary tree structure. Like RDR, MCRDR has a default root node which always returns a true to guarantee a conclusion with every input. However, unlike RDR's true and false branch at each parent node, MCRDR has exclusively exception (or true) branches. An MCRDR parent node can have any number of exception branches and each branch is followed when the parent node condition is true for a given case. Inferencing in MCRDR is generally similar to RDR except that if a condition is false at a parent node, then no child nodes will be evaluated. Another variation between the two methods is that unlike in RDR, a case in MCRDR can return multiple effective conclusions depending on how many of the terminating nodes evaluate to true for the case.

Inferencing in MCRDR is such that the root node is evaluated first, and then all nodes connected to the root are tested next. The nodes that evaluate to true will have their child nodes tested and the ripple continues until a terminating node is reached or until all children are false. The effective conclusions for the case will then be a collation of the firing terminal rules in each branch. The following diagram illustrates the inferencing process in an MCRDR structure. In the diagram, it is assumed the structure has an input of X= {b,d,f,k,o,h,e,m,t,y}.

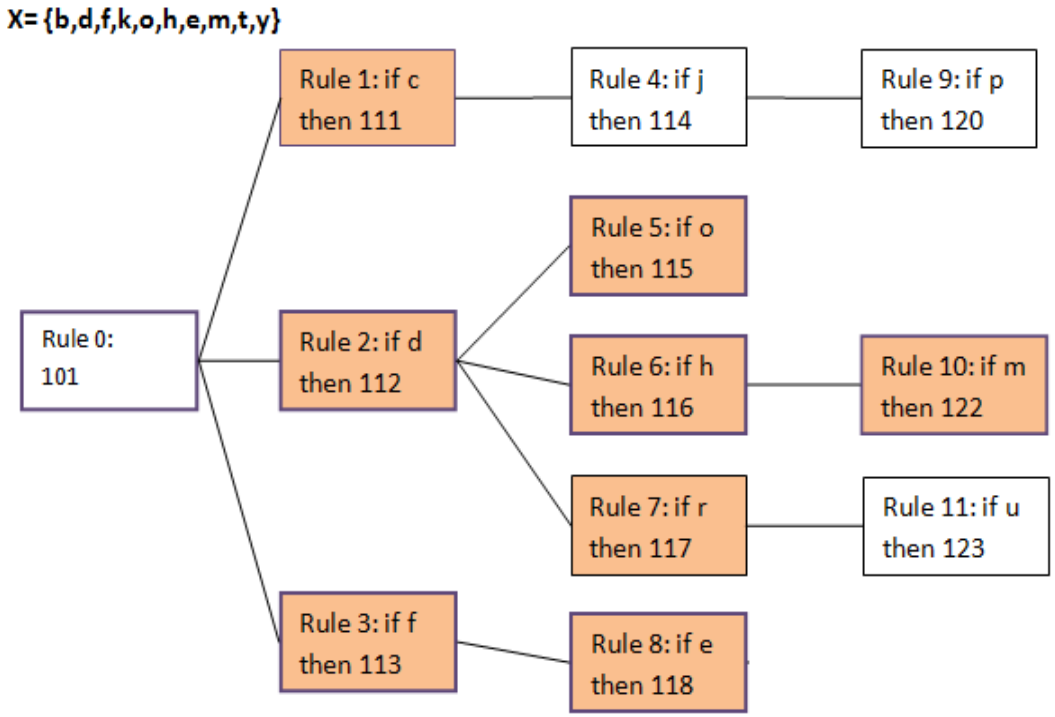

Figure 3-5. An MCRDR structure showing evaluated rules for the input X= {b, d, f, k, o, h, e, m, t, y}. Note that the default rule (Rule 0) always returns true.

For the example at Figure 3-5, the root node is evaluated first, returning true by default. Rules 1, 2, and 3 are evaluated next. Rule 1 returns false so its child is not checked. Rule 2 returns true so all its branches (Rules 5, 6 and 7) are checked. Rule 5 returns true and is a terminating rule. Rule 6 also returns true so Rule 10 is checked, which also returns true and terminates. Rule 7 returns false so its leaf node is not checked. Since Rule 3 returns true, its branch (Rule 8) is evaluated and returns true. Rule 8's child (Rule 12) is then evaluated and returns false. After all inferencing is done, the effective conclusions are 115, 122 and 118. These are the last true firing nodes in each independent branch.

3.7 MCRDR Learning

MCRDR learns as rules are added to its KB. When the expert considers an MCRDR classification to be incorrect, he/she supplies the correct classification and the system determines where a new rule to handle this case will be located. It is advised that the new rule cannot be simply assumed to be a refinement of the rule that gave the current conclusion as it may be a completely independent classification or the classification may be

just wrong (Kang, Compton, & Preston, 1995). For each new rule addition, the system has to determine if the given conclusion is wrong and not to be repeated, in which case a stopping rule is added. If the classification given by MCRDR was wrong and has to be stopped, then the new rule is added as a refinement to the old rule to prevent the wrong classification. This new rule is called the stopping rule (Kang, Compton, & Preston, 1995). The same approach is followed if the wrong classification is to be replaced by a new rule. If the system determines that the current conclusion should be replaced by another, then a refining rule is added. The new rule is added as a child of the terminating rule at the end of the path so that the wrong classification will not be given for this case again. Sometimes the current conclusion may not be wrong but an additional, independent conclusion may be needed. In this case, a new rule is added to the same node as the current node's parent so that the new rule adds an additional conclusion to the current one (Kang, Compton, & Preston, 1995). This allows for multiple conclusions for a given case because the newly added rule and the current rule will both be evaluated the next time the current case is fed into the inference engine again (Dazeley, 2007).

As in RDR, rule addition in MCRDR has to be done such that the new rule does not invalidate pre-existing rules and so that the new rule satisfies the current case (Kang, Compton, & Preston, 1995). In RDR, this validation process is done by comparing the current case with the cornerstone case. The same concept applies in MCRDR. When a rule is created, the relevant cornerstone cases and the current case will be used to formulate a difference list. A formula has been suggested by Kang, Compton, & Preston (1995) for selecting conditions from a difference list in a way that ensures the resulting rule is sufficiently precise. Given two cornerstones C and D and the current case E, at least condition should be selected from one of the sets dif_A or $diff_B$.

$$dif_A = E\ and\ !(C\ \cap D) \quad (3\text{-}1)$$

$$dif_B = \sum DL_i \quad (3\text{-}2)$$

Where E = exclusively E elements or conditions exclusively in the current case and DL_i is the difference between the current case and the *i*th cornerstone case.

In this situation, conditions for the new rule will be selected from comparing the difference list between case E and the common conditions (or intersection) of cornerstone cases C and D. For this scenario, consider the following example for current case E and cornerstone cases C and D:

E= { b, c }, C= { c, e, f} and *D= {d, f}*

dif_A = *E* and !*(C* ∩ *D) = {b}* and !*{ f}*

= b, !f

The condition for the new rule from the example above can then be *(if a* AND *! f)*. The above operations may sometimes return an empty list when using the approach defined in dif_A does not produce any set of conditions for the new rule (Kang, Compton, & Preston, 1995). In such circumstances, the rule conditions are formulated from comparisons involving the current case and a series of cornerstone cases. The final rule conditions are composed from a series of difference lists between the current case and each of the cornerstone cases shown in dif_B as defined at equation (3-2)

Consider a situation where current case *E= {a, d, g}* and the two cornerstone cases are *C= {a, d, q}* and *D= {g, k}.*

In this example, dif_A is empty because E is empty and $(C \cap D)$ is also empty.

$diff_B$ then becomes $DL_{E,C}$ + $DL_{E,D}$ where $DL_{E,C}$ is the difference list between case E and case D and $DL_{E,D}$ is the difference list between case E and case D.

So, dif_B *= (g, !q) + (a, d, !k)*

The new rule can be constructed using conditions from both $DL_{E,C}$ and $DL_{E,D}$. The new rule can also be uniquely identified by combining at least one attribute from the two lists (Kang, Compton, & Preston, 1995). For example, the rule could be (*if g AND !k*).

Despite its seemingly laborious process, rule addition in MCRDR is not as complicated in practice. This is because as soon as the expert selects an attribute from a difference list, other cornerstone cases are checked against this attribute and all cases not satisfied are removed from the list. Effectively, this elimination process reduces a significant amount of comparisons the expert has to do, resulting in an average of three difference lists and four comparisons per rule addition (Kang, Compton, & Preston, 1995). For example, using the Tic Tac Toe dataset (Aha, 1991), it was found that rule addition in MCRDR will involve at most seven cases and an average of 1.84 cases (Kang, Compton, & Preston, 1995). A single rule addition involving seven cases is rare, which is why the average for more than 110 rules is approximately two cases. Adding a single rule in MCRDR is reported to be about twice as long as in RDR (Kang, Compton, & Preston, 1995). Rule addition in RDR is relatively rapid and

trivial (Compton & Jansen, 1988; Kang, Compton, & Preston, 1995) , lasting not more than two minutes in a KB of 1000 rules and under five minutes for a KB with 10000 rules (Compton, Peters, Edwards, & Lavers, 2005). It must be noted also that in RDR, only one cornerstone case is reviewed per rule addition whereas in MCRDR, around three cornerstone cases are seen by the expert. Despite the multiple cornerstone cases, Kang, Compton, & Preston (1995) advise that rule addition in MCRDR will practically not be more than twice the time of RDR.

Alternative Incremental KA Approaches

The idea of incremental knowledge acquisition is not only applied within RDR methods. In fact, the incremental knowledge acquisition approach of RDR was introduced as advancement to existing methods. Some of the more recent work in non-RDR incremental KA includes a proposal for a context-aware system that uses a collaborative based KA approach (Joffe, Havakuk, Herskovic, Patel, & Bernstam, 2012). Joffe et al (2012) argue that the disadvantage with RDR-based KA is that it requires explicit expert input and that the proposed collaborative KA method could be used to implicitly collect knowledge from multiple experts unobtrusively. In this approach, knowledge was acquired by following and observing medical experts as they performed their duties. The system would then be incrementally expanded by iteration and corroboration of existing knowledge by newly collected data from other experts. The resulting knowledge base had a precision range of 0.8 to 1, justifying the viability of this approach (Joffe, Havakuk, Herskovic, Patel, & Bernstam, 2012).

Another non-RDR incremental KA approach features a combination of a Naïve-Bayes algorithm and a modified fuzzy partitioning method (Liu & Liang, 2011). In this system, a fuzzy unsupervised clustering procedure is combined with a Bayesian method into an incremental learning algorithm. The system was applied in text classification (where pre-labelled data is reportedly often difficult to obtain) and recorded good results with an average precision of over 90% in six different datasets (Liu & Liang, 2011). Other alternative incremental KA methods include Learn++, an incremental KA method comprising an ensemble of classifiers and a weighted output method (Polika, Udpa, Udpa, & Honavar, 2004). Another alternative at incremental KA is the Mixture of Experts (ME) system, consisting of several neural networks (or expert networks) (Ng, McLachlan, & Lee, 2006).

3.8 Prudence in Knowledge Bases

A major limitation of KBS not directly addressed by RDR nor simple MCRDR is brittleness. Brittleness is a common occurrence in expert systems where the system produces a conclusion that may not be correct and sometimes impractical. A classical example of brittleness is demonstrated by the widely cited 'pregnant male' phenomenon, where a chemical pathological system diagnosed a male as pregnant because they had a hormone secreting tumour, which caused a detection of a pregnancy hormone (Prayote, 2007). The ES in question did not know (and was not taught) the fact that only females can be pregnant. This occasional slip-up is common whenever some knowledge just outside of a system's expertise is needed (Dazeley, 2007). In this case, the pathological system needed to know that males cannot get pregnant.

One of the earliest implications from MCRDR noted by Kang, Compton, & Preston (1995) was the possibility to equip the MCRDR method with an ability to warn the system administrator whenever the method gives an erroneous conclusion. The first RDR based attempt to address brittleness in this way was a technique named Prudence (Edwards, Kang, Preston, & Compton, 1995). Prudence and credentials were introduced as properties of expert systems that managed how errors are managed in an ES. Credentials represented an ES's performance profile and would give the users a better understanding of the system's credibility (Edwards, Kang, Preston, & Compton, 1995). Prudence and Credentials were introduced through two approaches; Feature Exception Prudence (FEP) and Feature Recognition Prudence (FRP). In FEP, as the ES processes a case and produces a conclusion, the case's attributes are compared against processed cases. If any features of the current case are found to be unacceptable relative to other processed cases, they are flagged as exceptions. These exceptions are noted because they potentially invalidate a case's conclusion. FRP compares a case's rule path against other paths of the same conclusion. If a similar path could satisfy a different conclusion, then this indicates a possibility that the original conclusion might have been incorrect (Edwards, Kang, Preston, & Compton, 1995). The system had a high false positive rate but marked a breakthrough in the development of prudent expert system. This was a giant leap in ensuring that knowledge-based systems have some way of realising their limits and that even with insufficient expertise or missing information, the systems do not make claims such as the pregnant male example described earlier.

Another early attempt at prudence included keeping a record of all processed cases and issuing a warning each time a previously unseen case was processed (Compton, Preston, Edwards, & Kang, 1996). For numerical attributes, the system maintains a range of previously seen values. Each time a new minimum or maximum is seen, a flag is raised so that the conclusion is verified by the expert. For categorical attributes, an incoming value is compared with a list of previously seen values so that a warning is given if the value is new (Compton, Preston, Edwards, & Kang, 1996). It was concluded that this method's accuracy was not sufficient enough for real world use because of the high rate of false positives (Compton, Preston, Edwards, & Kang, 1996). Table 3.3 shows the performance summary of this method. In the table, False Positives (FP) are cases where a warning was produced unnecessarily. The False Negatives (FN) are cases where a warning was not given but should have been. The True Positives (TP) include cases where a warning was rightly issued. When no warning is required and none is issued, then the case is a True Negative (TN).

Dataset	FN %	TP %	TN %	FP %
Garvan	0.2	2.4	83	15
Chess	0.3	1.3	91	7
Tic Tac Toe	1.5	3.8	81	14

Table 3-3. Early prudence system's performance statistics

The prudence methods profiled above form a part of attribute based prudence methods (Dazeley & Kang, 2008; Dazeley, 2007). These methods rely on the presence or absence of case attributes to issue a warning (Dazeley & Kang, 2008). The attribute based group were later extended by a model based method; Ripple Down Models (Prayote, 2007), which will be examined closer in the next chapter. Another type of the prudence methods is the structural based prudence method which uses the paths followed by an inference process to determine if a warning should be issued or not (Dazeley & Kang, 2008). One of published and successful of these methods is Rated MCRDR (Dazeley & Kang, 2008), which will also be analysed in depth in the next chapter. Figure 3-6 illustrates the two types of prudence methods.

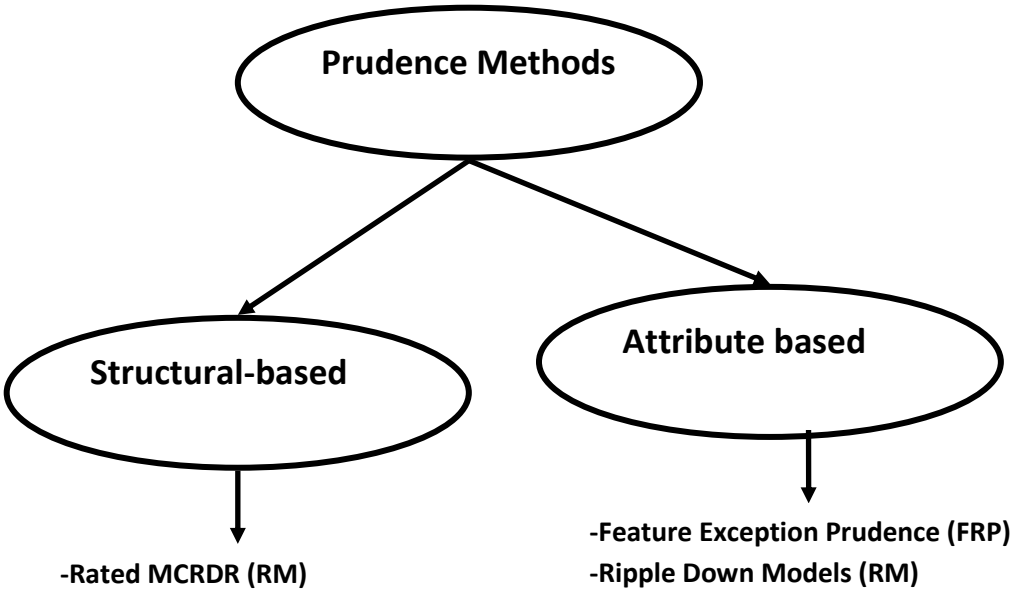

Figure 3-6. The two types of Prudence and their examples

It has been proposed that the reliance on attributes by attribute based methods (include model based methods) limit their applicability in domains with a controlled number of relevant attributes (Dazeley & Kang, 2008). On the other hand, it can be argued that the reliance of structural based prudence on a context drawn from the arrangement of fired rules or traversed paths during inferencing may not be valid for all domains. For this reason, this thesis combines attribute based and structural based prudence to form a new Integrated Prudence Analysis. This new method and its organisation will be discussed further in the next Chapter after an analysis of Ripple Down Models and Rated MCRDR.

3.9 Chapter Summary

This chapter introduced Knowledge-Based Systems (KBS) and Expert Systems (ES) and gave an analysis of components of KBS/ES. A few examples demonstrating the commercial worth of ES were also cited. The chapter introduced Rule-Based Systems (RBS) as a common variant of ES and listed some of the main limitations of knowledge acquisition and maintenance approaches used in conventional ES. As an alternative to these shortcomings, RDR was analysed in depth including how rules are added and how RDR learns. The Multiple

Classification version of RDR (MCRDR) was also discussed at length together with why RDR was redeveloped to MCRDR. Although innovative in its knowledge addition, contextual nature of rules and its maintenance approach, MCRDR was still found not to address the issue of brittleness in KBS. Prudence was introduced as the ES ability to notify the expert each time a new case or potentially wrong conclusion was produced. An overview of two early attempts at curbing brittleness was also given. The second prudence method forms the basis of RDM, a successful attribute based prudence method to be analysed further in the next chapter. Another successful method to be examined in the next chapter is RM, the only known structural based prudence method. This research proposes merging the two methods into an integrated prudence approach, also to be discussed at length in the next chapter. Chapter 4 examines RDM and RM in detail and describes Integrated Prudence Analysis, a merger of the two approaches.

4. Prudent RDR methods

4.1 Introduction

Following the previous chapter where RDR and prudence were introduced, this chapter analyses two of the reported successful prudence systems. The chapter starts with a description of the structure of Rated MCRDR (RM) and how it works. Using a similar format, the chapter then presents Ripple Down Models (RDM). Finally the chapter describes Integrated Prudence Analysis (IPA), a merger of RM and RDM. IPA is a novel method proposed and created by this research project. IPA is also one of the main contributions of this thesis. Three variations of IPA are described including a short explanation of the reasoning behind each proposed version.

4.2 Rated MCRDR

Rated MCRDR (RM) was founded on (Dazeley, 2007)'s proposal that a pattern of firing terminating rules in an MCRDR structure could reveal a hidden context that could be useful in understanding the KB's domain. In his PhD thesis, (Dazeley, 2007) explains that MCRDR had been identified by Gaines (2000) as a possible methodology for modelling the process of practice. Process of practice is defined as an instance when people are not compelled by logic or reflex but rather by their habits at the time (Dazeley, 2007). An example of such instances in official and business operations has been modelled through KBS (Gaines, 2000). Gaines (2000) further asserts that MCRDR is a promising method for modelling the concept of process of practice. It is on this idea that Dazeley (2007) founded Rated MCRDR.

Dazeley (2007) argues that although the advantages of multiple conclusions in MCRDR are obvious, there still could be further unexplored benefit in finding some context between the structure's individual paths. He further posits that there are correlations between inferencing paths and that these correlations could contribute to understanding more about the knowledge base's domain. A hybrid system, known as Rated MCRDR was suggested to determine these correlations and their usefulness in improving learning. RM combines MCRDR outputs with an Artificial Neural Network (ANN) to maximise the online learning ability of MCRDR with the generalisation ability of the ANN (Dazeley, 2007). The RM system

has 2 main components; the MCRDR engine and an ANN. The diagram below shows a schematic of the components of RM.

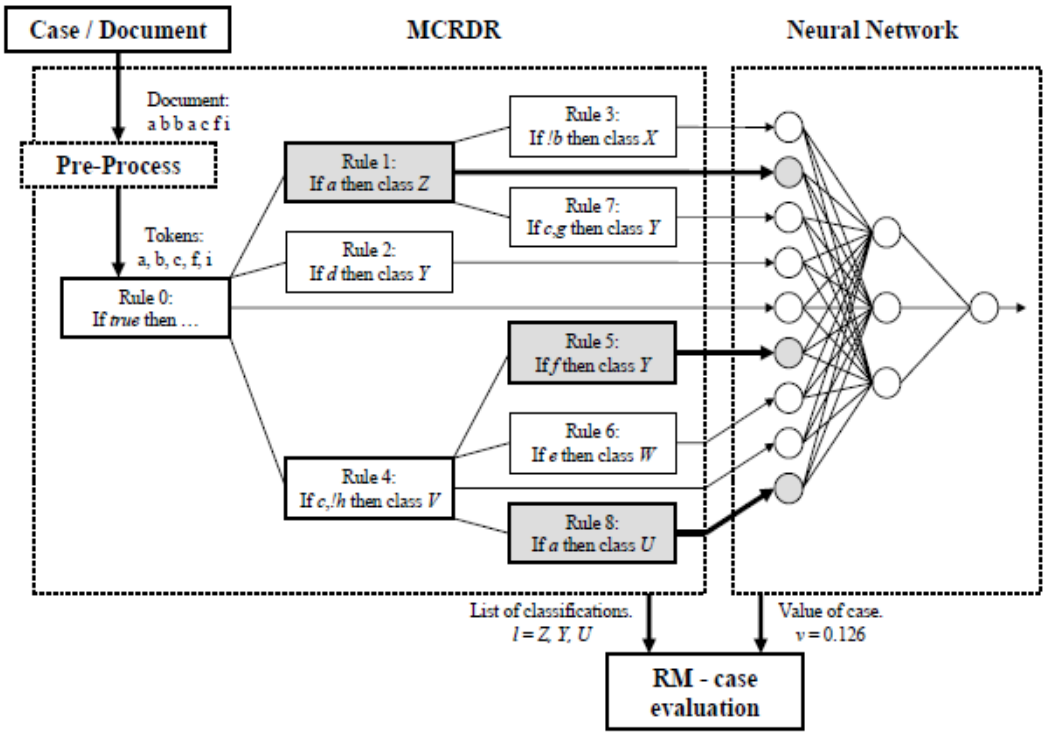

Figure 4-1. An overview of RM's main components. MCRDR output classifications are indexed and fed to an ANN and the two components outputs are evaluated. (Dazeley, 2007)

MCRDR Indexing

After a case is processed by the MCRDR engine, an indexing mechanism converts the MCRDR outputs into a set of binary inputs for the ANN. Five different methods of connecting the MCRDR outputs to the ANN were identified and tested. These included:

- Class Association (CA),
- Attribute Association (AA)
- Rule Path Association (RPA)
- Terminating Rule Association (TRA)
- Decreasing Rule Path Association (DRPA) method

In the CA association method, each possible classification for the domain is identified and assigned to a dedicated ANN input neuron. Each time the particular class is the final classification, the appropriate input neuron is turned on (assigned a 1 value), otherwise it is switched off (assigned a 0 value) (Dazeley, 2007). The advantage of the CA method is that

the input neuron size will never be more than k + 1, where k is the number of possible classifications for the domain. The extra input neuron in this case represents the bias neuron if the ANN uses a bias. The downside to the reduced input space also means that the network receives limited information about the case, consequently affecting the ANN's ability to generalize (Dazeley, 2007). This method was found to be less accurate than the other methods.

For the AA method, each attribute is associated to an input neuron and, as in the CA method, any attribute included in the firing terminating rule is switched on while non used attributes are switched off. With this method, the input layer can have up to k + 1 input neurons, where k is the number of case attributes. According to Dazeley (2007), the advantage with this method is that the system could still generalise well even if a rarely traversed MCRDR path involving similar attributes as in other rules is taken. The disadvantage with this method is that if the domain has contextually dependant attributes, the ANN inputs would fire globally, not exclusively when it was necessary (Dazeley, 2007). Like the CA, the AA method was also outperformed by other methods.

The RPA method associates each rule path to an ANN input neuron. As in the other methods, input neurons are switched on for every firing rule path. For this method, the network's size will be determined by the indexed rule paths. For example, for a dataset with k rule paths, the input size will have up to k + 1, where k is the number of indexed paths. The RPA method extracted a significant amount of information from the MCRDR engine and had a larger ANN input size than the other association methods. It was found to be one of the best methods and was used for the RM system developed in this project.

The TRA method is a modification of the RPA method where for every case the only indices switched on are terminal, firing rules (Dazeley, 2007). This method worked occasionally but does not pass on contextual content to the ANN. Dazeley (2007) suggests that it may be useful when individual rules represent a parent's specialization.

The DRPA method is a combination of the RPA and the TRA methods. The indexing of MCRDR outputs is as in RPA and TRA where each rule path is linked to a dedicated ANN input. In this method, the ANN input's value is determined by the last firing node's distance from the terminating rule. For example, if the last firing node for some rule is the second last after the terminating rule, then a typical value for this neuron can be: 1-0.25 = 0.75. If the last firing node is two nodes from the terminating rule, then the value for such a neuron is: 1-2(0.25) =

0.5. If the last firing node is the terminating node then the neuron value is 1 since the distance between the node and the terminal rule is 0. In the three examples, a distance step of 0.25 was chosen arbitrarily. Dazeley (2007) informs that the idea behind DRPA is to have a means of removing an input's discreetness relative to its closeness to the terminating rule. Although it occasionally outperformed the RPA method, the DRPA did not significantly aid the ANN in learning (Dazeley, 2007).

Artificial Neural Network

Artificial Neural Networks (ANN) are a biologically inspired form of distributed computing usually comprising a set of nodes (including input, hidden and output) and weighted connections between them (Chen, Hsu, & Shen, 2005). ANNs can also be defined as a topology formed by organizing nodes into layers and linking the layers of neurons. The nodes are interconnected by weighted connections, and the weights are adjusted when data is presented to the network during a training process (Dayhoff & DeLeo, 2001). ANNs provide a mapping from the input space to the output space so can learn from the given cases and generalize the internal patterns of a given data set (Guo & Li, 2008). They adapt the connection weights between neurons and approximate a mapping function that models the provided training data (Marsland, 2003). Learning in supervised ANNs is usually achieved through two phases: the feed-forward and back-propagation (BP) processes.

In a typical feed-forward process, the sum of the products of the input nodes and their weights are passed through a threshold and the result at each hidden node is multiplied by the corresponding weights (hidden-to-output node connections). The sum of each connection and hidden node product is collected at each output node, passed through a continuously differentiable non-liner function (CDNLF) and the ANN output is determined. Usually, the sigmoid function (CDNLF) is used at the hidden and output nodes. The sigmoid function provides enough information about the output to earlier nodes (hidden and input) so that the weights can be adjusted accordingly to reduce the difference between the ANN's calculated output and the desired target output (Beale & Jackson, 1991). The sigmoid function is given in equation (4-1).

$$f(net) = 1/(1 + e^{-k\,net}) \quad (4\text{-}1)$$

Where k is a positive constant and controls the breadth of the function.

Figure 4-2 illustrates a simple 3 layered artificial neural network.

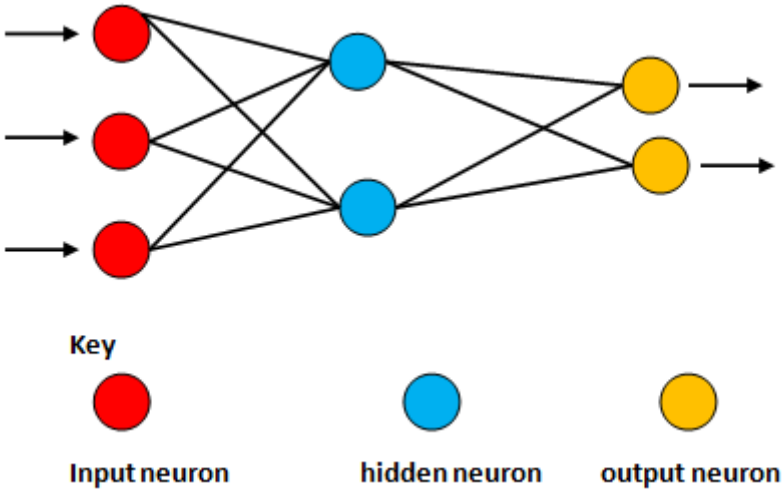

Figure 4-2. A simple three layered ANN with input, hidden and output layers.

A number of variations of neural networks with modified algorithms are in use today in different applications, including in fraud detection. The use of ANNs in fraud detection spans almost all major forms of fraud including telecommunications fraud, financial fraud and computer intrusion fraud among others (Kou , Lu, Sirongwatana, & Huang, 2004). In RM, an ANN is used as a secondary classifier where MCRDR outputs are indexed and passed to the ANN and the output is used to substantiate the MCRDR conclusion. The next section describes the different RM versions and the variation of ANN used in each configuration.

RM Versions

Up to seven different function fitting methods were tested in different versions of the RM system (Dazeley, 2007). In the end, it was decided that the best configuration of RM comprised the ANN as the complementary component. This section gives a summary of some of the versions of RM and why the ANN was ultimately selected as the better component for the MCRDR outputs.

RM Weighted (RM_w)

RM_w does not use an ANN but assigns a value to an MCRDR rule, class or attribute depending on the association method. After a case is processed, the values are averaged and the returned value represents the effective output of RM_w . When a new feature is identified, it is assigned a value using equation 4-2 (Dazeley, 2007).

$$w_n = \frac{z(v_R - v_{RM})}{m} \quad (4\text{-}2)$$

Where w_n is the new feature's value, v_R is the system's value of the reward, v_{RM} is the effective average of the existing features and z is the step modifier, determining the degree of adjustment for the new features.

Although RMw was introduced as a benchmark for other RM versions, it was found to be incapable of the discovery of any non linear relationships nor any useful form of learning (Dazeley, 2007).

RM with Basic Linear Technique (RM_{lr})

RM_{lr} uses a single layer ANN whose new inputs are initialised randomly to capture linear relationships. The use of a single layer ANN allows for the addition of new nodes with little consideration for their initial weights. The method allowed the network to appropriately adjust its weight when a new neuron was connected and to generalise the linear relationships between features. The RM_{lr} method was expected to only generalise linearly separable relationships between features. It was also found to be better than the previously discussed RM_w (Dazeley, 2007).

RM with Advanced Linear Technique (RM_{IA})

RM_{IA} is similar to RM_{lr} except that after a new node is added, a single step delta initialisation (ssdi) is used to set the node's weight instead of a random value (Dazeley, 2007). The ssdi formula enables the network to find the weight required to instantly produce the right value. When calculating the new weight, an inverse of the sigmoid may be used to determine the target output (T_{ws}) because the feed-forward process uses a sigmoid function at the output node. Alternatively, the T_{ws} can just be plugged in if known *apriori*. The aim of the ssdi is to assign the required weight to the connections so that the T_{ws} is produced by the network. To do this, the network error at the output node (E_o) has to be known because the sum of the output error and the actual output (net_o) equals the T_{ws}. So,

$$E_o = T_{ws} - net_o \quad (4\text{-}3)$$

The network's actual output (net_o) is calculated during the feed-forward operation. Equation (4-4) shows, in a simple format how this value is calculated.

$$net_o = \sum_i^n (x_i w_{io}) \quad (4\text{-}4)$$

Where x_i is the input value at neuron *i* and w_{io} is the weight value for the connection between the input and output neuron.

The weight value (w_n) for each new connection is then determined by dividing the network error by the number of new inputs *k*. Equation (4-5) captures this step.

$$w_n = z\left(\frac{E_o}{k}\right) \quad (4\text{-}5)$$

Where z is the step modifier and determines the degree of precision of the new weight.

As stated earlier, the target output can be determined by reversing the sigmoid thresholding operation that was done at the feed-forward stage. The in T_{ws} in this case is determined by finding the inverse of the sigmoid, as shown in Equation (4-6)

$$T_{ws} = \log((f_{net} + \partial + 0.5)/(0.5 - f_{net} + \partial))/k \quad (4\text{-}6)$$

Where f_{net} is the feed-forward value after the sigmoid thresholding operation.

The detailed ssdi for a linear network then becomes:

$$w_n = z(\ (log((f_{net} + \partial + 0.5)/(0.5 - f_{net} + \partial))/k) - \ (\sum_i^n (x_i w_{io}))/k) \quad (4\text{-}7)$$

Where E_o is represented by T_{ws} as shown in Equation (4-6) and net_o shown in Equation (4-4).

This system was found to be able to generalise the correct output immediately after receiving expert knowledge, although this capability was restricted exclusively to linear relationships (Dazeley, 2007).

RM with Basic Non-Linear Technique (RM_{bp})

To improve on the linear techniques, a system using a multi layered neural network with a back-propagation algorithm was introduced. RM_{bp} uses a symmetric sigmoid threshold function for the feed-forward process and assigns random weights to new input nodes (Dazeley, 2007). When new hidden nodes are added, all input-to-hidden nodes (IH) connections are assigned a small random value, as are the new hidden-to-output node (HO) connections. Although at a slow rate, the RM_{bp} system was expected to learn non linear relationships.

RM with Advanced Non-Linear Technique (RM_{bpA})

To address the limitations of the other versions, RM_{bpA} has a number of features that enable it to learn faster, generalise non linear relationships and do so without losing already learned information. To conserve currently learned content, the system employs shortcut connections from a new input neuron directly to each output node (Dazeley, 2007). The new connections are used to support the weight adjustments needed to produce a particular output. With the shortcut mechanism, the network is capable of adjusting immediately when a new input is added to the network while still retaining previously added information. Dazeley (2007) adds that with the shortcut connections, there are effectively two networks; the multilayer perceptron for non-linear relationships and the single layer network (shortcut connections) for learning linear relationships. Since the ANN inputs are directly connected to the indexed MCRDR conclusions, when a new conclusion fires in the MCRDR structure, a corresponding input will have to be introduced to the ANN. When such a new input node is added to the ANN, the following adjustments are made depending on the state of the network at the time and the settings for the hidden layer size.

- If a new input is added and no new hidden neurons have to be added; connections are added from the new input node to all hidden nodes. The weights of these connections are initialised to zero. A new shortcut connection is also added from the new input neuron to all output nodes. The weight for the new input-to output node (IO) connection is calculated using the modified ssdi equation for a for a non linear network with a hidden layer. The ssdi* is similar in concept to the ssdi equation (explained earlier) except that the network's actual output (net_o) include the weighted sum from the shortcut (IO) connections (Dazeley, 2007). So for a non linear network,

$$net_o = (\sum_i^n x_i w_{io}) + (\sum_h^q x_h w_{ho}) \quad (4\text{-}8)$$

Where q is the number of hidden nodes and x_h is the non-linear output at the hidden node h.

The new ssdi* then becomes,

$$w_n = z(T_{ws} - \left((\sum_i^n x_i w_{io}) + (\sum_h^q x_h w_{ho})\right)/k) \quad (4\text{-}9)$$

Where T_{ws} is as shown in Equation (4-6).

The diagram at Figure 4-3 shows the connections and weight allocations when adding a new input and no hidden neurons to a network.

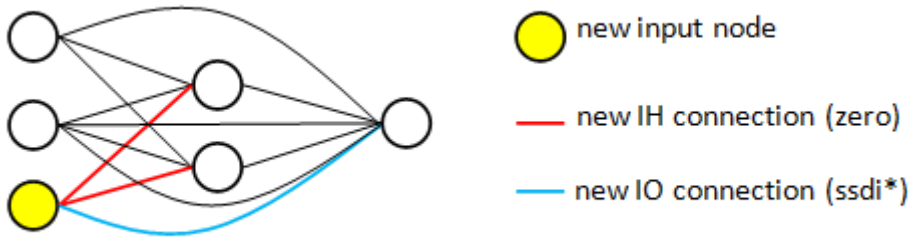

Figure 4-3. Dynamic addition of a new input node to an ANN.

- If a new input is added and a new hidden neuron has to be added; new IH connections are added from the new input neurons to the old hidden neurons. These connections are initialised to zero so that they have no immediate influence on existing generalisations (Dazeley, 2007). New IH connections from all input neurons to the new hidden neurons are also added and initialised by random numbers. New HO neurons from the new hidden neurons to the output neurons. The connections are initialised to zero (Dazeley, 2007). Finally, new shortcut (IO) connections are added from the new input neurons to all output neurons and their initial values calculated using the ssdi equation for a non linear network shown at Equation (4-9). Figure 4-4 gives a pictorial abstraction of this process.

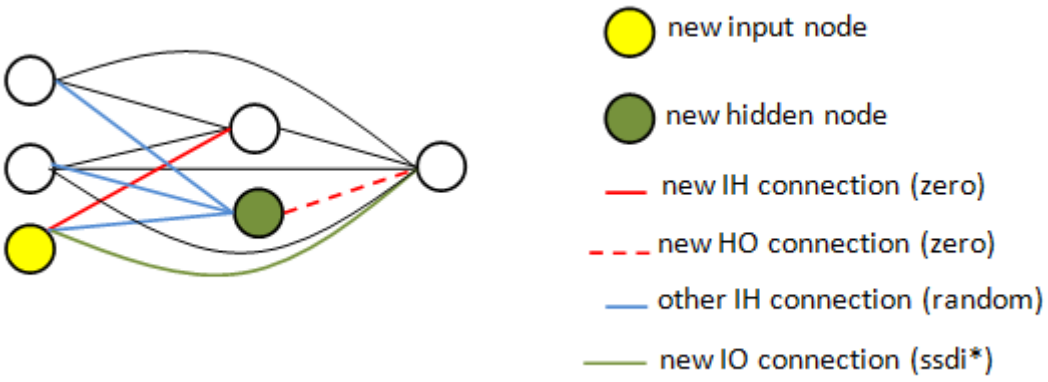

Figure 4-4. Adding a new input and hidden node to an ANN

The RM_{bpA} system is an advancement on all the other previously described systems in that it learns both linear and non-linear solutions effectively and immediately and can do so without disturbing previously learned information (Dazeley, 2007). However, Dazeley (2007) cautions that the system may struggle to generalise in a domain where only a few exclusively excitatory or exclusively inhibitory inputs fire. This is because currently, the system works well when a combination of both excitatory and inhibitory; and a considerably larger size of inputs actually fire (Dazeley, 2007).

RM with Radial Basis Function (RM_{rbf})

Developed incrementally like the RM_{bpA}, the RM_{rbf} system was first developed with a basic Radial Basis Function (RBF) then later improved to use the Advanced RBF. The earlier version was shown to have a good online learning ability and possibly prevent over-fitting by only adding hidden neurons when new input nodes are added. The drawback with this system was that hidden nodes could not be added for unique patterns and this had a significant impact on the system's ability to learn (Dazeley, 2007). The latter version, using an advanced RBF was aimed at addressing the limitations of the basic version. The new version was altered to occasionally add new hidden nodes even when there was no input, thereby achieving a faster and better generalisation. One of the main disadvantages of this system is that it was reliant on many parameters, many of which were directly related to the system's performance.

Comparing RM Versions

Given the range of techniques applied in the RM systems above, a series of tests were performed by Dazeley (2007) to determine the best system. The systems expected to perform better overall are the latter versions of the non-linear methods and the advanced RBF method. This is because each of these systems is a focused re-development of the earlier versions addressing specific limitations in these versions. The tests were categorised into two tasks: classification and prediction. Classification measures the system's ability to correctly identify a case's class and prediction determines the system's ability to group similar cases as well as identify a new class and create a new rule (Dazeley, 2007).

In the classification tests, RM_{bpA} and RM_{rbf} outperformed the other methods across five datasets (Dazeley, 2007). This was not unexpected, since the advantages of these systems over the others were obvious. Between the two, RM_{bpA} consistently outperformed RM_{rbf}, although the latter system marginally posted better results in training (Dazeley, 2007). For

prediction, only one dataset was used because the C4.5 simulated experts for the other datasets could not be used to determine a form of classification and a class value. Again, it was shown that RM_{bpA} and RM_{rbf} were the clear favourites. However, this time, (Dazeley, 2007) advises that the RBF method may be better for simpler domains with linear relationships and that RM_{bpA} would be ideal for more complex domains. Overall, (Dazeley, 2007) informs that although they occasionally performed well, the RBF methods were volatile, inconsistent and dependant on the order of the cases.

Generally, RM_{bpA} with the RPA indexing showed better results in a consistent fashion and is consequently the better RM version. Additional tests comparing RM_{bpA} and an ANN on its own further proved RM_{bpA}'s superiority as the method repeatedly outperformed the ANN in all the datasets (Dazeley, 2007). Informed by the methods, tests and results described in the previous and current sections, this research adopted the RM_{bpA} system as the best RM version and used this configuration for all tests and evaluations used in this project.

4.3 Ripple Down Models

RDR's inferencing process also partitions a search space into smaller sub-regions (Prayote, 2007). It is on this principle that RDM prudence is primarily based. RDM was originally developed to be a network intrusion detection system. The system depends on RDR's inferencing processes in which new partitions are created as new rules are added. Exploiting this particular feature of RDR and some others, RDM was designed to be an online ID system which would detect outliers as the system learnt the network behaviour (Prayote, 2007). The system assumes that the partitions resulting from RDR inferencing are homogenous and that the data within each partition is uniformly distributed (Prayote, 2007). The RDM system can be dissected into three main components; RDR modelling, Outlier estimation and Final RDR classification. The diagram at Figure 4-5 shows a simplified representation of RDM with its main components.

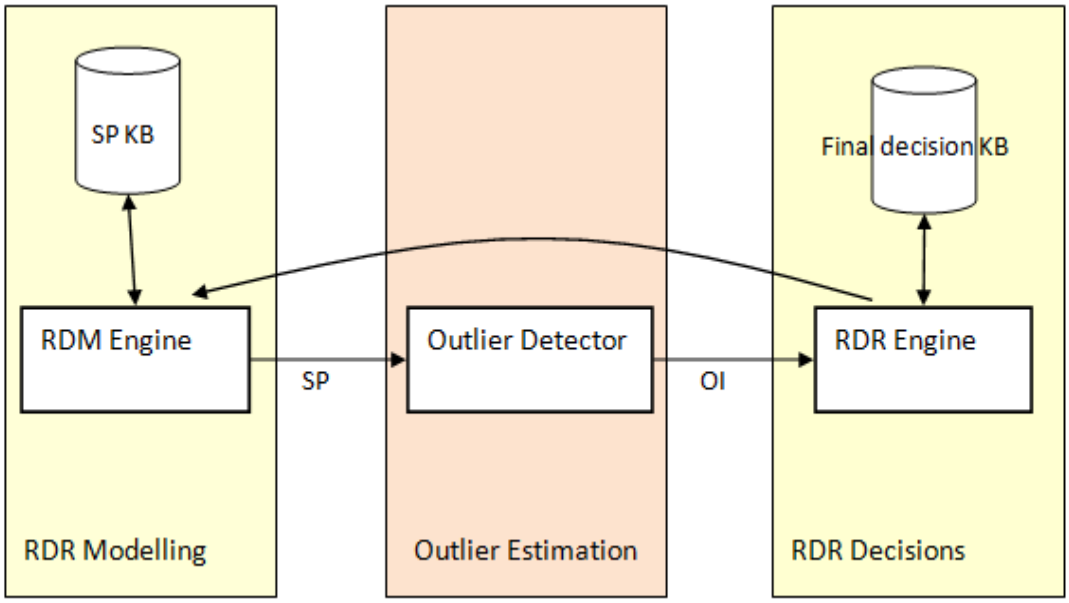

Figure 4-5. Main components of RDM. The RDM Engine processes a case and passes an SP to the Outlier Detector which passes Outlier Indexes to the RDR Engine for final classifications.

A case is introduced to the system through the RDR modelling component. In this phase, the RDR engine retrieves an appropriate Situated Profile (SP) for the case. As in RDR, there is a default SP in case none of the available SPs match the current case. The SP is then passed to the outlier estimation components where each attribute value is searched for possible outliers. The outlier detection results are passed to the RDR decision base where another RDR inferencing process takes place to classify the case as either an anomaly or not. The RDR classification is first confirmed with an expert and is stored in the RDR decision base if correct. Otherwise, a new SP and a new rule are created for the case (Prayote, 2007). The next sections explain RDM's three components in detail.

RDR Modelling

In conventional RDR, after a case is processed, a corresponding conclusion or class is returned (see chapter 3 for inferencing in RDR). In RDM, after RDR inferencing, a model is returned instead of a class. This model, also called a Situated Profile, contains profiles describing each of the attributes matching the current case. For this reason, the method is known as the Ripple Down Models. The diagram below shows the main disparity between a traditional RDR inference process and an RDM inference process.

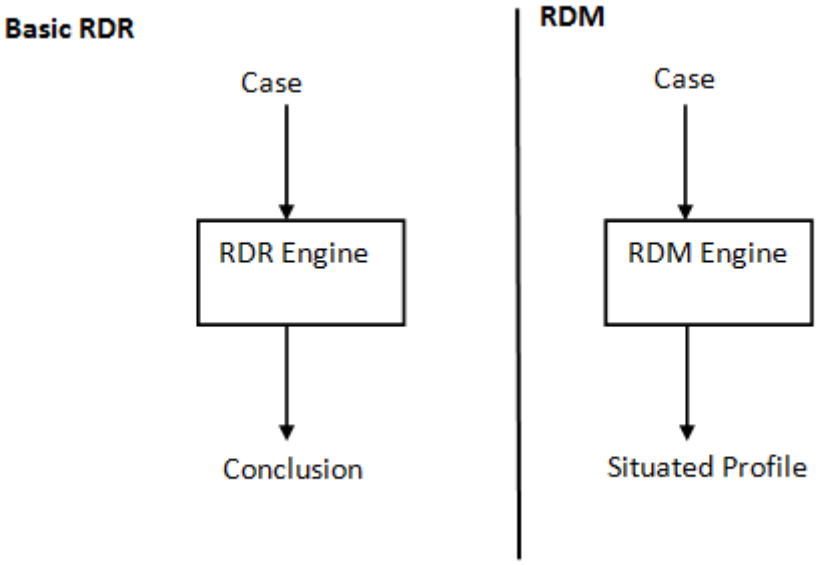

Figure 4-6. Output of an RDR inference Engine and an RDM inference engine

Each SP is made up of the same number of Profiles as the case attributes. If a dataset has *n* attributes, then each SP will contain *n* Profiles. The use of Profiles is adopted from general ID systems' Profile modelling classification where a system either defines a normal behaviour Profile (Anomaly Detection) or maintains a list of known unacceptable Profiles (Signature Detection). A Profile in the RDM context is a behaviour pattern, representing a homogenous subspace in the domain data. A homogeneous region of data according to Prayote (2007)'s model is uniformly distributed on an interval *[a, b]* where the probability density function *f(x)* and the cumulative density function *D(x)* are defined further in the following sections.

For example, in Prayote (2007)'s application, a Profile consisted of a homogenous pattern of network traffic and each Profile represented a particular situation in the network. In his network traffic AD system, Prayote (2007) states that the network traffic rises to 5MB per minute for 30 minutes on weekdays from 6am and that this could be seen as the "weekday-6am" Profile (Prayote, 2007). Specifically, each RDM SP was defined by the TimeOfDay, DayOfWeek and Season Profiles (Prayote, 2007). As is typical in RDR rule addition, a new SP is added whenever a new situation is identified. If the current case corresponds to a pre existing situation, a matching SP is retrieved using the RDR inferencing process (Prayote, 2007). The retrieved SP is then passed to the Outlier Estimation module where each Profile is screened for anomalies. The next section explains this process.

Outlier Estimation in RDM

The SP passed from the RDM Modelling component is analysed by an outlier detection algorithm to detect outliers within Profiles. Each Profile is screened by one of two algorithms and a binary indication of whether the case's attribute value for that Profile is an outlier or not is returned. If a Profile models a categorical attribute, the Outlier Estimation for Categorical Attributes (OECA) algorithm is used to detect if the case's value for the attribute is an outlier or not. For numerical Profiles, the Outlier Estimation with Backward Adaptability (OEBA) method is applied and a binary outlier index is returned for that Profile. For each SP of *n* Profiles, the Outlier Detector (OEBA and OECA) returns a set of *n* binary indexes indicating the status of each of the case's attributes. OEBA and OECA are described further below.

Outlier Estimation with Backward Adaptability (OEBA)

The OEBA algorithm is probability dependant and aims to model a continuous attribute in a dynamic environment (Prayote, 2007). The accuracy of OEBA is reliant on the homogeneity of a subspace, so the algorithm assumes the following:

1. All attributes are independent of each other.
2. Each cluster is homogenous and the data within each region behaves similarly.
3. Each homogenous region is uniformly distributed on the interval [a, b] and,

$$
\begin{aligned}
f(x) &= 0 && \text{for } x < a \\
&= 1/(b-a) && \text{for } a \le x \le b \\
&= 0 && \text{for } x > b
\end{aligned}
$$

And

$$
\begin{aligned}
D(x) &= 0 && \text{for } x < a \\
&= (x-a)/(b-a) && \text{for } a \le x \le b \\
&= 1 && \text{for } x > b
\end{aligned}
$$

Where f(x) is the probability density function and D(x) is the cumulative distributive function (Prayote, 2007).

Given these assumptions, the probability of a value falling into a range [c, d] inside the interval [a, b] is,

$$P(c \leq x \leq d) = D(d) - D(c)$$
$$= (d-a)/(b-a) - (c-a)/(b-a)$$
$$(d-c)/(b-a)$$

Where $c \geq a$ and $d \leq b$.

The function $P(c \leq x \leq d)$ defines the probability of a given range [c, d] for a single point. Finding the Range Probability (RP^k) for k consecutive independent points, $P(c \leq x_1 \leq d)$, $P(c \leq x_2 \leq d)$ and $P(c \leq x_k \leq d)$ in the region [c, d] within the interval [a, b] can be calculated as follows:

$$RP^k = P(c \leq x_1 \leq d) \times P(c \leq x_2 \leq d) \times P(c \leq x_k \leq d) \quad (4\text{-}9)$$
$$= (\beta/\alpha)^k$$

Where $\beta = (d-c)$ and $\alpha = (b-a)$.

Prayote (2007) advises that any point x that lies beyond this interval is not necessarily an outlier as it could be a new, previously unseen point. Similarly, it is possible for an outlier to be within this region. In a case where a point identified as an outlier is indeed within the region, the interval [a, b] can be re-defined as [a, x] if $x \geq b$ or [x, b] if $x \leq a$. The RP^k for k objects falling within the sub-region [a, b] is now $(\beta/\alpha)^k$ where $\alpha = (x-a)$ and $x \geq b$ or $(\beta/\alpha)^k$ where $\alpha = (b-x)$ and $x \leq a$. The Outlier Estimation algorithm pseudo code based on these assumptions is given below:

Outlier Estimation

```
Let a: the minimum of the range after n observations
b: maximum of the range after n observations
x: new observation
T: the least confidence level at which x is accepted as a
value of the population
q: binary indication (0 if not outlier, 1 if outlier)
q=0
If x ≥ a and  x ≤ b then
    n= n + 1,
else if x > b and RP^k ≥  T then
    b=x, n= n+1
else if  x < a and  RP^k ≥  T then
    a=x, n= n+ 1
else
    q=1 //outlier
end if
```

After many cases have been observed, outliers are likely to stand out as extremes (Prayote, 2007). After each Outlier Estimation update, the Backward Adaptability (BA) mechanism inspects the updated range and resets the Profile's maximum or minimum value depending on where the outlier was detected. For example, if an outlier was accepted into the model as an upper bound, the following cases would all fall within a sub-range such that the previous bound will be identified as an outlier. The BA method would then correct the range accordingly. The BA pseudo code is given below:

Backward Adaptability

```
Let a: the minimum of the range after n observations
b: maximum of the range after n observations
x: new observation
T: the least confidence level at which x is accepted as a
value of the population
c: min(x, observed minimum since a was set)
d: max(x, observed maximum since b was set)
ka: population within the range after a was set
kb: population within the range since b was set
if RP^k(a, d)< T then
    b= d, kb =0
end if
if RP^k(c, b)< T then
    a=c, ka= 0
end if
```

Outlier Estimation for Categorical Attributes (OECA)

The OECA algorithm is used to detect outliers in categorical Profiles. The algorithm is modelled on the following assumptions:

1. Each attribute is independent of another.
2. Each attribute is uniformly distributed. This means that the occurrence probability of attribute values is equal and observes the function,

$$f(x) = \frac{1}{v} \quad (4\text{-}10)$$

Where v is the number of different observed value.

From these assumptions and the geometric distribution function, (Prayote, 2007) deduces that the probability of a new value B being seen after k trials where A was seen is,

$$f(x) = \left(\frac{1}{v+1}\right)\left(1 - \frac{1}{v+1}\right)^k \quad (4\text{-}11)$$

A measure M is used to specify how well the cases match the Profile. M= 1.0 for seen values and when a value is new, then M is the probability of having this value after v different values from k observations. This statement is simplified in equation (4-12).

M(x) = 1.0 If x has previously been observed

And (4-12)

M(x) = $\left(\frac{1}{v+1}\right)(1-\frac{1}{v+1})^{k}$ Otherwise

Prayote (2007) suggests that a new value after a long spell of previously seen values is more likely to be an outlier than when new values have been regularly observed. To incorporate this aspect, the probability of a new value (M) is compared with the probability of the last value to be accepted into a Profile (M_A). The ratio of these probabilities is called as the New Value Ratio (NVR) and is given as,

$$NVR = \frac{M}{M_A} \quad (4\text{-}13)$$

OECA uses NVR to justify whether a value is an outlier or not. The higher the NVR, the less likely it is to be an outlier (Prayote, 2007). If a value is less than a given Threshold, it is marked as an outlier. The OECA algorithm pseudo-code is shown below:

Outlier Estimation for Categorical Attributes

```
Let x: an observation
T: Threshold
q: binary indication (0 if not outlier, 1 if outlier)
If M(x)= 1.0 OR
If NVR > T then
   q=0
else
   q=1
end if
```

RDR Final Decisions

After the Outlier Detection methods process the SP, a set of binary outlier indexes representing each Profile is passed to the RDR knowledge base for a final classification of the case. The re-classification of a case is done because OEBA and OECA only assess each attribute in isolation and have no way of conclusively determining if a case is an anomaly or not. Prayote (2007) explains that although each case may have more than one attribute, some individual attributes may be more important than others and this is why an additional

RDR inferencing will give an expert a chance to justify a conclusion based on particular attributes. The second RDR knowledge base also allows the expert to confirm or correct OEBA/OECA results, hence reducing false positives. If a conclusion was found to be wrong, the expert defines a new situation and a new SP and corresponding rule are added to the appropriate KBs (Prayote, 2007).

4.4 MCRDR based RDM

The RDM system described earlier uses a single classification RDR method in both the Situated Profile KB and the final decisions KB. This research introduced two main changes to the approach used by (Prayote, 2007). First, a multiple classification RDR KB was used for the Situated Profiles. This was done primarily to enable the system to handle both single classification and multiple classification problems. The other rationale for using an MCRDR-SP-KB was informed by (Richards, 2009)'s work showing that even for single classification domains, MCRDR produced a more compact KB with fewer redundancies than single class RDR.

The second modification to the original RDM was the use of OEBA/OECA as a complementary classifier to MCRDR rather than as a preliminary outlier detector incapable of classifying a case. In Prayote (2007)'s design, OEBA/OECA's output is further processed by an RDR engine where the final decisions are made. In this project, a set of binary indexes for each SP were summed and if the aggregated outlier index was above some threshold, the case was classified as an anomaly. The outlier detection methods (OECA/OEBA) were used in a similar manner as the ANN in RM, where the systems effectively produce two classifications; the MCRDR classification and the outlier detector (ANN/OEBA /OECA) classification.

The primary difference between single class RDM and multiple classification RDM is the MCRDR knowledge engine. Instead of producing a single conclusion for any input as in single Class RDM, the multiple classification RDM knowledge is capable of multiple conclusions hence multiple Situated Profiles if necessary. The modification to change single class RDM to MC-RDM was hence enabled through the use of a MCRDR knowledge engine, developed as part of the Ballarat Incremental Knowledge Engine (BIKE) project (Dazeley, Warner, Johnson, & Vamplew, 2010). With an MCRDR engine, the SP-KB could then be modified to produce more than one SP at a time when needed. The second section of chapter 6 (i.e. 6.2) provides

results on the comparisons of Prayote (2007)'s single classification RDM and the multiple classification RDM developed by this project.

4.5 Integrated Prudence Analysis

Integrated Prudence Analysis (IPA) is a novel prudence method proposed, developed and evaluated in this thesis. IPA is an attempt to combine an attribute based prudence method (MC-RDM) and a structural based prudence method (RM). Attribute based prudence techniques' warning mechanisms are dependent on the presence or absence of particular values of case attributes. For a warning to be issued, a case's attributes are compared to similar cases in the KB and if the case is determined to have (or miss) the necessary attributes then a warning is given (Dazeley & Kang, 2008). Structural based prudence learns the paths traversed during inferencing and decides whether a warning is necessary or not given a path's novelty and consistency with previously traversed paths.

The combination of these approaches is anticipated to take advantage of the supplementary rule path context extraction of RM and partition based outlier detection methods of MC-RDM. It is hoped that combining the principally opposite methods will eliminate the inherent limitations of using each method individually. RM's impressive results are dependent on the method's extraction of hidden contexts within the inference rule paths which enhance a classifier's learning of a domain. Similarly, the segmentation of a domain into homogeneous partitions before applying an outlier detection algorithm gives MC-RDM its performance advantage. The strength of IPA, is therefore anticipated to be in the unison of the two methods rather than in some other configuration. This project proposes three combination strategies of RM and MC-RDM; IPA_{OR}, IPA_{AND} and IPA_{ANN}. The diagram at Figure 4-7 illustrates an overview of the IPA_{OR} /IPA_{AND} architecture.

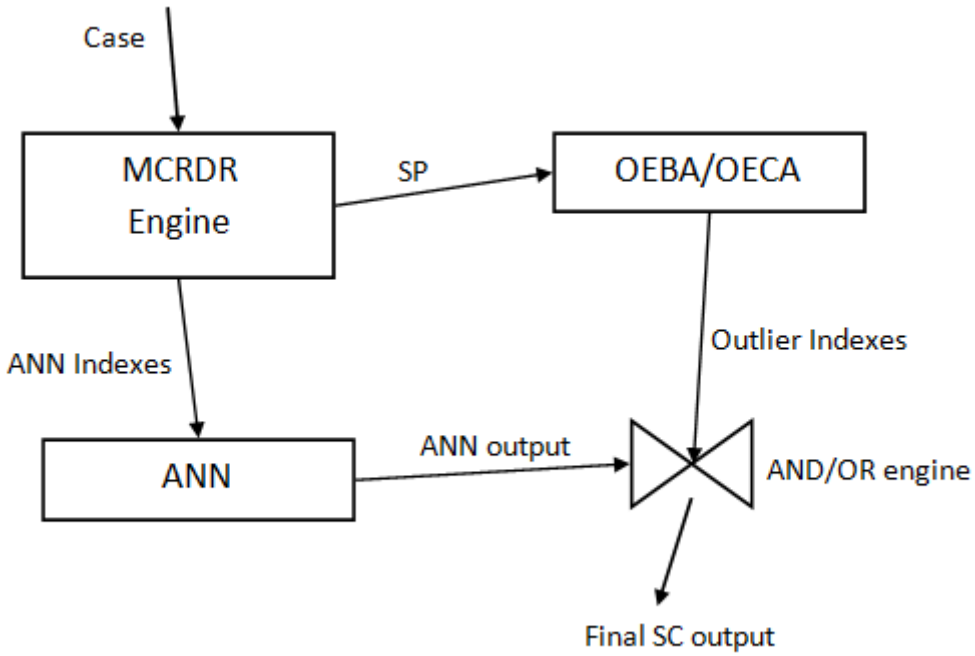

Figure 4-7. The IPA_{OR} /IPA_{AND} system. The final Secondary Classifier output is either the AND or OR result of the ANN output and the aggregated Outlier Index from OEBA/OECA.

The IPA_{OR} /IPA_{AND} is a fairly simple method, combining RM's ANN output with the MC-RDM's aggregated outlier index through an AND or OR connection. The method is a generic connection of RDM and MC-RDM to an MCRDR engine. The slight difference in IPA_{OR} /IPA_{AND} is that a single MCRDR engine serves both the OEBA/OECA outlier detectors and the ANN. The indexing of rule paths (for the ANN) and the creation of SP's (for the outlier detectors) is generated from a single MCRDR engine. Each time a new MCRDR rule is added, a new SP is created and a new ANN index is generated. The effective output of the complementary classifier can either be the AND or OR result of the ANN index and the OEBA/OECA aggregated outlier index.

The IPA_{ANN} is slightly more complicated than the IPA_{OR} /IPA_{AND}. In IPA_{ANN}, the aggregated outlier index from the MC-RDM outlier detectors is combined with the MCRDR indexes and fed into the ANN. The idea behind this combination was to see if feeding the MCRDR hidden context (indexed MCRDR outputs) and the MC-RDM aggregated outlier index straight into an ANN would result in the ANN being a better classifier than the ANN in RM or the MC-RDM outlier detectors. The only difference between the two ANNs is that the ANN in the IPA_{ANN} takes the aggregated outlier index as its additional input. Other settings and configurations are as described in RDM and RM. Figure 4-8 shows the structure of the IPA_{ANN}.

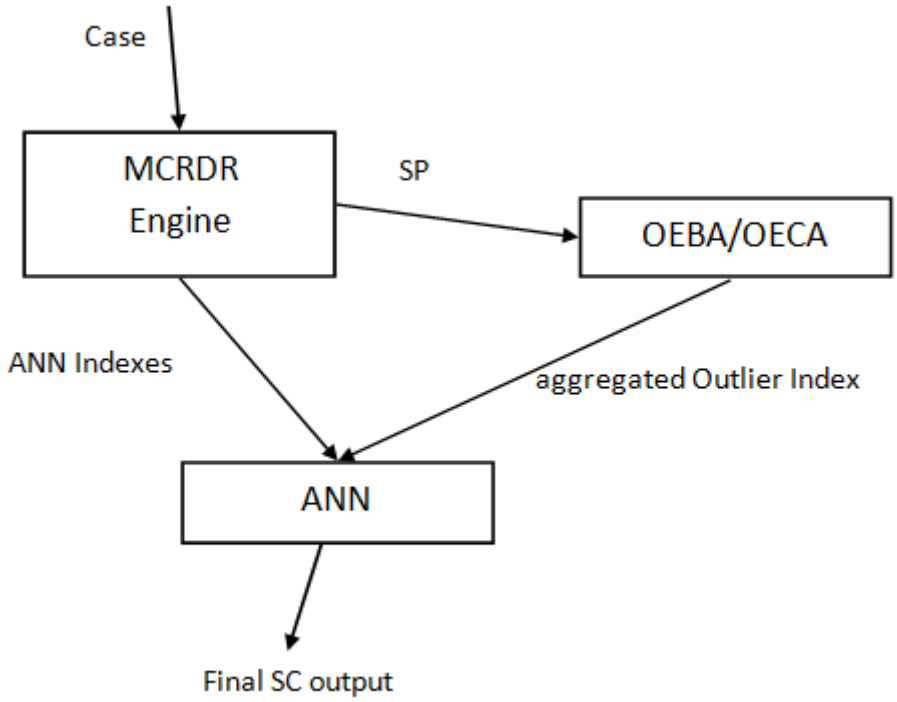

Figure 4-8. A generic overview of the IPA_{ANN} system. In this approach, the ANN input combines output from the rule path indexes and the aggregated outlier indexes from OEBA/OECA.

In RM, the ANN's input exclusively comprises MCRDR rule path indexes. In IPA_{ANN}, these indexes are added to the aggregated outlier index from MC-RDM's outlier detectors. The additional input from MC-RDM could boost the ANN's classification and learning ability. The indexing of MCRDR rule paths and generation of SP's in IPA_{ANN} is the same as in IPA_{OR} /IPA_{AND}. The differentiating feature of IPA_{ANN} is that the ANN input includes MCRDR rule path indexes and MC-RDM's aggregated outlier index.

Combining RM and RDM is expected to leverage the strengths of RM and RDM and thereby eliminate some of the systems' individual vulnerabilities. The strategic combination is also anticipated to take advantage of the supplementary rule path context extraction of RM and partition based outlier detection methods of MC-RDM. The ANN component is meant to extract any additional context there may be from paths of fired MCRDR rules while the OEBA and OECA components screen attribute inconsistencies to detect probable outliers. These strengths, combined with other inherent advantages of the RDR methodology over conventional knowledge bases is justify IPA as a potentially strong system. The three IPA versions were tested across eight public and private datasets and a full analysis of the tests and evaluations is discussed in chapter 6.

4.6 Chapter Summary

This chapter presented a detailed analysis of the two successful prudence methods to date; RM and RDM. The structure and operation of each method was also explained in full including the redevelopment of the original single class RDM to a multiple classification RDM or MC-RDM. The chapter concludes with an introduction and analysis of IPA, a new integrated method combining elements of both RM and RDM. IPA combines two different approaches to prudence; structural based approaches and attribute based approaches. The different variations of IPA were given along with what each variation was intended to achieve. The next chapter introduces and describes the overall methodology applied in this project including the evaluation metrics and the datasets used to compare and rate the three systems. The chapter explains the need for simulated experts and how they were used to test the three systems.

5. Methodology

5.1 Introduction

The previous chapter gave a detailed analysis of RM and RDM and introduced IPA as a merger of the two prudent methods. The different configurations of IPA were also discussed at length. This chapter explains the methodology used to evaluate and compare the three prudent RDR methods. The chapter surveys a few general KBS evaluation approaches highlighting the strengths and limitations of each approach. In the later sections of the chapter, the discussion is narrowed down to specific RDR evaluations with a focus on how they will be applied in this project. The chapter explains the use of Simulated Experts to enable faster, repeatable tests. A full description of the datasets used in this project is also given. The chapter then explains the specific evaluation metrics used in this project: class accuracy and prudence accuracy and their relevance in assessing and comparing prudent RDR methods.

5.2 The Need for KBS evaluation

Knowledge-Based Systems (KBS), like other systems need to be evaluated before deployment into the systems' intended domain. A 1993 journal article advised that at that time KBS were being evaluated at three critical points: during design to measure the degree of performance at a particular stage of development; after a KBS has been completed to match the desired behaviour with the actual behaviour; and also when different implementations are proposed to compare the two methods (Guida & Mauri, 1993). A technical report published the same year (Grogono, Preece, Shingal, & Suen, 1993) summarised that evaluation is done primarily to check if the KBS does what it is meant to do. Over five years later, a research paper published a KBS evaluation approach whose objective was to improve the system's performance by revealing its strengths and weaknesses (Lippmann, et al., 2000).

There is a range of specific and organisation-relevant purposes for evaluating a KBS but usually the greater objective includes determination of the system's actual performance against the specified performance. Similarly, there exists a variety of evaluation methods for

KBS, the algorithms they employ and their Knowledge Acquisition (KA) techniques. Some of these evaluation approaches are discussed in the following section.

5.3 Evaluation Metrics for KBS

Apparently, a weakness in some evaluation methods of KBS is that they do not give an explicit measure of a KBS's power and complexity. Furthermore, these methods are rarely packaged into independently usable packages (Salim, Villavicencio, & Timmerman, 2003). A journal article published in 2003 (Salim, Villavicencio, & Timmerman, 2003) applies the Function Point Analysis (FPA) method as an evaluation basis for comparing KBS for use in industrial applications. FPA is conventionally used to measure the accuracy and user friendliness of a software package. In FPA, specific features of a software application are assigned numerical values representing their degree of merit or lack-of. Each feature's value is then multiplied by an assigned weight and the sum of all features' values is obtained and scaled to a small range, usually 0-5 (Salim, Villavicencio, & Timmerman, 2003). This value corresponds to the software package's level of accuracy and user friendliness. An illustration on the use of the FPA's direct method is shown in the following example.

To complete the assessment the questions shown at Figure 5-1 are answered within an appropriate category shown in the assessment matrix at Table 5-1 (Salim, Villavicencio, & Timmerman, 2003).

1. Is there enough information to evaluate the software?
2. Does the software give the same answer as other methods?
3. Does the software give the same answer as a human?
4. Does the software provide the right answer for the right reasons?
5. Is the software accurate in its answer(s)?
6. Is the answer complete? Does the user need to do additional work to get a usable result?
7. Is the procedure of getting the answer simple and clear?
8. Does the answer change if new but irrelevant data is entered into the software?
9. Can the system clearly explain its reasoning technique to the user?
10. Does the system require a lot of irrelevant questions to get to reach the answer?
11. Does the answer change if irrelevant changes are made to the system's rules?
12. Does the software crash or hang up in its host computer?
13. Does the system give warnings for cases involving incomplete data or rules?
14. Is the cost of the system justified by its performance?
15. Does the shell have all the features listed on the vendor's literature?
16. Does the software still provide answers with incomplete knowledge?
17. Can limitations of the shell be detected at this point in time?
18. Does the shell allow the user to expand a program if needed?
19. Can the system learn from increased data or experience?

Figure 5-1. FPA questions to determine accuracy and user-friendliness of a Knowledge-based system

Each True or False question listed above is represented by its corresponding number in Table 5-1. Using these questions, the matrix in the table is completed. The user (assessor) assigns a value between 0 and 5 in the Assessment column of each corresponding question. In this metric, 5 represents True and 0 represents False and the user can assign any number within the range depending on their assessment of the system. The product of the specified weights and the assessment level are then calculated for each row. The values shown in the table are an assessment for the MP2 software as evaluated in Salim, Villavicencio & Timmerman, (2003).

Category	Question	Assessment (A)	Weight (W)	W x A
Answer Correctness	1	5	2	10
	2	5	2	10
	3	5	2	10
	4	5	2	10
Answer Accuracy	5	5	2	10
	6	5	2	10
	7	5	2	10
Reasoning Technique Correctness	8	5	1	5
	9	3	1	3
	10	5	1	5
Sensitivity	11	5	1	5
Reliability	12	5	1	5
	13	5	1	5
Cost Effectiveness	14	5	1	5
	15	5	1	5
	16	1	1	1
Limitations	17	5	1	5
	18	4	1	4
	19	0	1	0
Results	Sum(W x A)/ 26			4.54

Table 5-1. FPA assessment matrix

The final rating ranges from 0 to 5, where 0 represents an unsatisfied user and 5 corresponds to a satisfied user. The MP2 software according to this assessment is reasonably accurate and quite user friendly as proven by its overall assessment level.

The FPA approach as applied in this context is subjective and cannot guarantee consistency of the final assessment rating. However, the weighted averaging provides an effective mechanism of smoothing differences between users. For example, if a different user totally disagreed that the procedure of getting the answer was simple and clear, and consequently awarded a value of 1 instead of 5 for question 7, this would result in an assessment sum of 110, a reduction of 8 from 118. In the final rating the new assessment translates to a rating of 4.23, a drop of 0.3 from the original rating. On the issue of subjectivity, Salim, Villavicencio, & Timmerman (2003) counter that the results still provide a useful scale for

comparing KBS and that the usefulness of the approach in comparing two relevant KBS is justified. For example, in a situation where a company may want to place more priority on a particular aspect of the KBS (e.g. cost effectiveness), the relevant metric in Table 5-1 (cost effectiveness) will be allocated an appropriate weight to reflect the respective significance attached to the metric.

Evaluating KBS involves two main activities according to Grogono et al (1993); verification and validation (V & V). Verification detects internal inconsistencies in the Knowledge Base (KB) and Validation checks that the system's actual behaviour is as specified in the design. In Grogono et al (1993)'s approach, evaluation is divided into two aspects, where verification checks for the KB's compliance with certain syntactic principles and validation checks if the system satisfies the specification requirements. The expertise in a KB may not always be perfect. In some case, a KB may contain omissions[1], unwilling misrepresentations or errors of the expertise. Verification in Grogono et al (1993)'s approach checks for these omissions. There are four kinds of omissions identified: ambivalence, circularity, redundancy and deficiency.

Ambivalence in a KB indicates a semantic constraint within a set of inferred final conclusions. For example, given that in a knowledge-base, conclusions {D, F} are a semantic constraint. If input A leads to conclusions B, D and F produced by rules 1, 3 and 4, then these rules are ambivalent because some elements of the output {B, D, F} are semantic constraints. This usually results from incompatible views by experts or mistakes during KA. Circularity exists in the KB if for some condition there is an indefinite firing of the rules. Grogono et al (1993) adds that although it indicates a problem with the KB, some inference engines are able to avoid rules that cause indefinite looping. Redundancy occurs when the exclusion of a rule from a KB has no effect on the conclusions of the inference engine. Although inefficient at times, redundancy is often accepted if the redundant rule does not contradict another. A KB is called deficient when no conclusion is made for a particular condition. Conventionally, a KB should be able to produce a conclusion from every input. The conclusion may be wrong, or the input may be beyond the KB's current expertise but some conclusion is expected all the same. The verification part of Grogono et al (1993)'s evaluation approach checks for these omissions and alerts the knowledge engineer.

[1] Omission is used instead of 'anomaly' in the original text to avoid confusion with anomaly as described in Chapter 2.

Validation ensures that the KBS complies with the specification requirements (Grogono et al, 1993). In most cases, this involves comparing the system with a human expert. Grogono et al (1993) concurs with the general notion in KBS literature (Gupta, 1991) that validation is two-fold; comprising Laboratory and Field validation. Laboratory validation measures the system's performance in a development environment, with developers supplying test cases (Grogono et al, 1993). Such tests are usually conducted to refine the system for deployment. According to (Grogono et al, 1993), Laboratory validation usually reveals KB related problems. Field validation is conducted with real users who were not involved in the development of the system. Usually, tests are conducted on a synthetic problem although in some cases, real problems are used. Most problems revealed in Field evaluation are interface-related (Grogono et al, 1993).

The V & V approach of Grogono et al (1993) presents a good approach to checking and ensuring the quality of a KBS. Validation uncovers undesirable omissions within the KB and Verification checks if the system works according to plan. The obvious limitation with this approach is its lack of an aggregate rating to indicate a degree of quality or satisfaction. Consequently, this approach may not be suitable for comparing KBS or determining which KBS is better than another.

Lippmann et al (2000) introduces Receiver Operating Characteristics (ROC) as an evaluation mechanism for Intrusion Detection (ID) systems. As detailed in Chapter 2, most ID systems are some type of a KBS. In Lippmann et al (2000)'s paper, the six ID systems used in the evaluation include three KBS. This effectively means that the ROC method was used to evaluate KBS and this is what earns the method its discussion in this context. The ROC method analyses differences between a KBS false alarm and detection rates and have traditionally been used as a way of visualising classifiers' performances (Fawcett, 2003). Some of the applications of ROC graphs are in signal detection theory to show classifiers' hit and false alarm rates; they are also used in the medical industry for diagnostic testing (Sweets, Dawes, & Monahan, 2000; Centor, 1991). The ROC analysis has been used in machine learning from as early as 1989 when ROC curves were used to evaluate algorithms (Spackman, 1989).

In Lippmann et al (2000)'s ROC approach, a threshold is determined to distinguish the detection accuracy and false alarm rates of acceptable and 'unusable' systems. For these particular tests, systems with more than 100 false alarms per day were regarded as unusable. To conduct the evaluation, KBS were initially provided with training data for seven

weeks. Each day, the systems received a dataset of around 287000 cases. After that, the systems were tested for two weeks and their detection accuracies and false alarm rates recorded. ROC curves were then plotted for each system, showing the correct detection rate versus the number of false alarms per day. The diagram at Figure 5-2 shows a sample ROC curve from one of Lippmann et al (2000)'s evaluations.

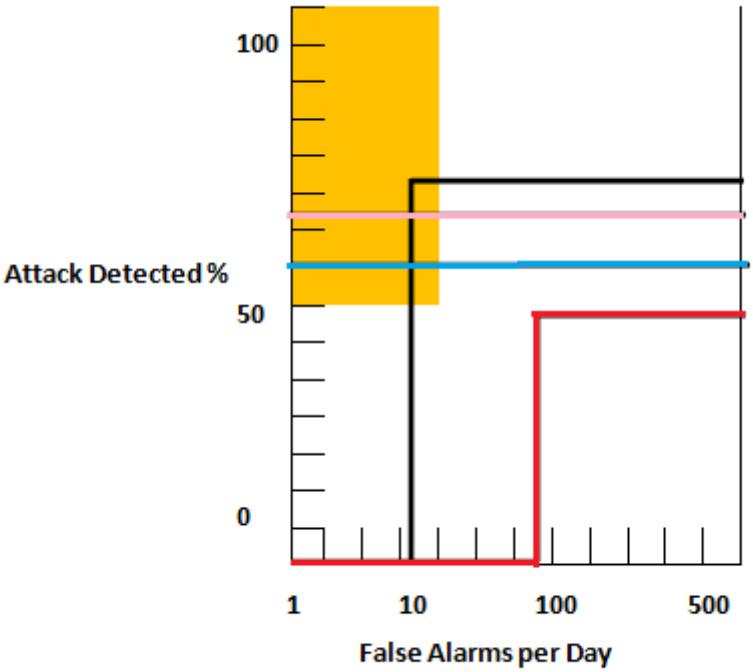

Figure 5-2. ROC curves for 4 systems, 3 of which are KBS. The highlighted area represents high detection rate and low false alarms.

The highlighted region (top left part of graph) corresponds to a detection rate of 50% or more and around 10 false alarms or less per day. Systems whose ROC curves pass through this region were considered good for the evaluations conducted in Lippmann et al (2000). Although Lippmann et al (2000) advise that this particular evaluation is not to compare individual systems, it can be inferred that the best system in this scheme would have its curve furthest to the left and closest to the top. The method has an added advantage of not being exclusively applicable to a single class of systems. However, the applicability of ROC to KBS evaluation may be limited by its disregard for KB integrity checking mechanisms, such as verification in Grogono et al (1993)'s approach. Despite that fact the Lippmann et al (2000) advises against it, ROC could be a good approach to comparing individual KBS accuracy and false alarm rates.

The evaluation method of Gholamreza & Schnabl (1997) presents a slight deviation from the approaches analysed in the previous sections. Instead of evaluating the KBS directly, Gholamreza & Schnabl (1997)'s method determines the worth of the KBS by evaluating the Data Mining (DM) algorithm(s) used in the system. DM is the process of identifying valid, potential useful patterns (in data) which can be used to predict future outcomes (Senator, 2009). DM is a component of Machine Learning and enables the discovery of pattern, models and relationships; collectively known as knowledge (Mihaela, 2006). Some of the techniques of DM include decision tree induction, rule induction, ANN, Support Vector Machines and many more (Mihaela, 2006). Some of these techniques are used in KBS such as the ones developed in this project. In this context, this is why evaluating these techniques is as essential as evaluating the KBS on which they are applied.

The method proposed by Gholamreza & Schnabl (1997) extends the concept of Data Envelopment Analysis (DEA) for application in evaluating DM algorithms. Gholamreza & Schnabl (1997) highlight a lack of objective metrics for evaluating both positive and negative characteristics of a DM algorithm. The DEA variant suggested by Gholamreza & Schnabl (1997) combines a KBS algorithm's advantages and disadvantages into an evaluation metric. The DEA concept was adapted from Charnes, Cooper & Rhodes (1978) and was originally used to develop a ranking system for Decision Making Units (DMU). In Gholamreza & Schnabl (1997)'s approach, each DMU is represented by a DM algorithm. In DEA, an algorithm's positive property (advantage/strength) is called an output component and a negative property (weakness) is called an input component (Gholamreza & Schnabl, 1997). For example, a typical output component can be the algorithm's accuracy rate and an input component can be the algorithm's training time. Output components generally have higher values and input components have lower values (Gholamreza & Schnabl, 1997). The efficiency of the algorithm (and indirectly the KBS) is then defined as the sum of weighted output components over the sum of weighted input components, represented mathematically in Equation 5-1.

$$eff = \sum O_w / \sum I_w \qquad (5\text{-}1)$$

where O_w is a weighted output component and I_w is a weighted input component.

The determination process of the component's weights is not systematic and is determined by the evaluator at the time (Gholamreza & Schnabl, 1997). This is a limitation acknowledged by Gholamreza & Schnabl (1997) who add that it is often difficult to award

objective weights. The most efficient algorithm according to DEA is one with the higher efficiency. The inclusion of a KBS algorithm's strengths and weaknesses into a combined efficiency metric is a good approach to comparing algorithms. Although there is a way around it, the subjectivity of components weight distribution remains a drawback to this approach as used in Gholamreza & Schnabl (1997).

A host of other approaches to evaluate a range of KBS in different application domains exist. Some evaluations are specific to a domain and some are generic and can be customised to any area of application. For example, Guo, Pan, & Heflin (2004) propose an evaluation scheme for KBS in large Web Ontology Language (OWL) applications using a modified version of the Leigh University Benchmark (Guo, Heflin, & Pan, 2003). In Guo, Heflin & Pan (2003)'s tests, four KBS were evaluated according to an extended model of the Leigh University Benchmark. However, the method does not provide a metric to rank or compare one KBS from another. Another application-specific evaluation approach for KBS is briefly discussed by Senator (2009). The approach is not methodically elaborated but briefly discussed as a list of metrics essential in evaluating DM applications in security KBS. Some of these metrics include undetected attacks and wrongly classified non-attacks. Senator (2009) adds that other metrics should consider whether the system is better detecting major threats frequently or minor threats less frequently. Another class of metrics should determine the rate of detection of new threats (Senator, 2009).

All of the evaluation methods presented above are useful in some aspect. Each approach has a particular strength based on the aspect of evaluation it targets such as the KBS algorithm, KB and output. Obviously each evaluation approach suits the KBS it was designed for (or tested on), so it may be hard to adapt some approaches to some other KBS. It would be desirable to combine some of these methods into a single evaluation approach that compares and ranks KBS based on multiple aspects including KB integrity, algorithm, output accuracy, usability and other important features. This method can then be packaged into a flexible software that can handle a wide range of KBS as advised by Salim et al (2003).

5.4 Evaluation in RDR KBS

The RDR methodology eliminates most of the limitations of other Knowledge Acquisition (KA) methods such that some evaluation methods targeting these limitations in conventional KBS are not relevant for RDR KBS. Consequently, most RDR evaluations (Dazeley, 2007),

(Prayote, 2007; Dazeley, Park, & Kang, 2011) have consistently used a simulated setup. The idea of simulating expertise to test RDR KB's stems from the fact that recruiting human experts for the same job would be both costly and time consuming (Compton, Preston, & Kang, 1995). Also, a simulated environment would allow for multiple, faster and more easily controllable tests (Compton & Cao, 2006). A Simulated Expert (SE) is a KBS used as a source of expertise in assessing a KA tool. In RDR, the KA method being trained or assessed acquires its knowledge from the SE (Compton, Preston, & Kang, 1995). As highlighted, the main advantage of simulated expertise is faster and flexible development and evaluation of KA methods (Compton, Preston, & Kang, 1995). The other advantage to use simulated expertise is the possibility of repeating tests and the consistency of simulated experts compared to human experts. Although machine learning derived rule traces do not strictly resemble human expertise (Cao & Compton, 2005), the ability to control variables and complete multiple tests in a short time gives SE a reasonable advantage and provides a good alternative to human expertise.

One of the latest methods of evaluating RDR based KBS is by Beydoun and Hoffman (2013) and Finlayson and Compton (2013). Beydoun and Hoffman (2013) propose an approach that essentially integrates knowledge acquisition with evaluation. In this, approach, instead of evaluating the KBS against a set of test cases, the knowledge acquisition process is monitored and the effectiveness of newly added rules is determined using a statistical analysis method (Beydoun & Hoffman, 2013). The statistical analysis component tracks the key parameters during knowledge acquisition and evaluates the coverage and accuracy of newly added rules. The proposed approach replaces the traditional knowledge evaluation task after a KBS is developed by a knowledge tracking during knowledge acquisition. The approach by Finlayson and Compton (2013) involves a separate KBS being built simultaneously with the RDR KBS but using a different learning technique. Each time the two knowledge bases differ on a case, then an expert checks each of the KBSs and makes relevant corrections (Finlayson & Compton, 2013).

The idea of run-time validation proposed by Finlayson and Compton (2013) is based on the premise that as KBS gets more complex, it is unlikely that the KBS will be completely validated against all cases. Beydoun and Hoffman (2013) argue that the proposed approach would require less training and testing data than the simulated expert and that rather than leaving evaluation to the end, the approach combines maintenance with evaluation. The paper asserts that the new approach may be potentially cheaper because an expert would

only be engaged when is strictly necessary. Although the idea seems viable and potentially effective, it was not applied to this research for two main reasons; first, one of the primary contributions of this research was to comprehensively evaluate RM and RDM using a conventional and familiar RDR approach. Using the new approach would have weakened the relevance of these comparisons relative to other past RDR performance evaluations. Secondly, the proposed method was published when the rigorous testing process of this research had already been completed. Repeating evaluations using this approach would have been a massive task and would have taken much longer to complete all tests. A simulated expert was therefore used for all evaluations in this project.

When building a KBS, a SE can be used as a source of expertise for the new KBS. With RDR being an incremental knowledge addition method, the new KBS will typically only have a default rule at the start and will incrementally add new rules (from the SE) to match incoming data. The default rule will be returned initially when the KBS has no other rules and later on when none of the available rules match the current case. When a case is introduced to the new KBS, the default rule fires and returns some dummy conclusion. Meanwhile, the same case is fed to the SE, which will return the correct conclusion. The new KBS's conclusion is then compared to the SE's conclusion. If the two conclusions match, then the new KBS is assumed to have the right rule for the case. If the conclusions do not match, the rule(s) fired by the SE are added to the new KBS to correspond to the current case. The process is repeated until the new KBS matches the SE or until the new KBS has learnt all the rules from the SE. This will obviously depend on whether the training data covers all the rules in the SE. An abstracted process for developing a new KBS using simulated expertise is shown below, adapted from Cao & Compton (2005).

Algorithm 5-1

1. Accept a new case.
2. Evaluate case against new knowledge base.
3. Evaluate case against SE and get a rule trace.
4. If the KBS conclusion does not match SE conclusion, then add rule (or rules) to new KB to correspond to new case
5. Go to step 1 for next case.

Simulated expertise can also be used in a similar manner to evaluate the accuracy of a KBS after its development. The difference in using an SE in evaluation is that there are no rule

additions to the assessed KB; instead, the assessed KB's conclusion is classified as either a True Positive or True Negative if the conclusion matches the SE's conclusion. When the SE and KBS have contradictory conclusions, the KBS's conclusion is classified as either True or False classification. The following process summarises the generic steps involved in evaluating a KBS using a SE. The generalised pseudo-code given below applies for both binary and non binary datasets.

Algorithm 5-2

1. Accept a new case.
2. Evaluate case against assessed KB.
3. Evaluate case against SE.
4. If the case is correctly classified (If SE conclusion matches KBS):
   - Increment True Classifications (TC)
5. If the case is not correctly classified (If SE conclusion does not match KBS):
   - Increment False Classifications (FC)
6. Go to step 1 for new case.

To evaluate the accuracy of a KBS during development, the following process is used. This same generic process was used in evaluating the RM, RDM and IPA systems discussed in depth in chapter 4. The process is used in the online evaluation of other RDR KBS (Compton, Preston, & Kang, 1995).

Algorithm 5-3

1. Accept a new case.
2. Evaluate case against assessed KB.
3. Evaluate case against SE.
4. If the case is correctly classified (If SE conclusion matches KBS):
   - Increment True Classifications (TC)
5. If the case is not correctly classified (If SE conclusion does not match KBS):
   - Increment False Classifications (FC)
   - Add rule (or rules) to new KB to correspond to new case
6. Go to step 1 for new case.

5.5 Creating a Simulated Expert

Usually, a Simulated Expert represents a complete version of the KB being built or a version faultless enough to be used as a benchmark against another knowledge-base system. Most RDR SE's were built by induction from a range of datasets using a learning algorithm including C4.5, InductRDR and See5 (Compton, Preston, & Kang, 1995; Maruatona, Vamplew, & Dazeley, 2012). For each SE, a corresponding dataset is first run on a machine learning algorithm to get a set of induction rules or a decision tree covering the dataset. To ensure that the SE is perfect and does not introduce conflicting rules into the KBS, any cases incorrectly classified in the machine learning decision tree or rule set can be removed so that every case in the dataset is correctly classified and covered by at least one rule in the rule set.

Another approach to specifying levels of expertise in a SE is detailed in (Compton, Preston, & Kang, 1995) where the smartest expertise involves selecting the top four conditions from the intersection between the SE rule traces and the difference list for the current case. The next level of expertise involves choosing a single condition from the intersection of the SE rules and the case's difference list and the lowest level of expertise (Dumb expert) selects all conditions from the case's difference list without reference to the SE's rules (Compton, Preston, & Kang, 1995). The use of varying levels of expertise in Compton et al (1995)'s work was to determine the effect of smart and dumb experts on the size and accuracy of the KBS being built. It was found that dumb expertise resulted in larger knowledge bases than smart experts and that RDR knowledge bases were not necessarily larger than machine learning knowledge bases as had been initially thought. The results also affirmed initial propositions that the smart expert produced a system with fewer errors than the dumb expert.

For this project, SEs of varying levels of expertise were built from See5 (Rulequest, 2012) decision trees. Some SEs were constructed from induction rules but were found to be slower and generally less accurate than their decision tree based counterparts. An induction rule SE took longer to add a rule to a developing KBS because each SE rule condition was compared to a case's difference list before the final rule could be constructed and added to the developing KBS. If only a few of the SE's rule conditions were compared to the difference list, the added rules tended to be more general and required the addition of more rules and eventually resulted in the same general rule firing for many cases, sometimes wrongly. In the end, the use of induction rules was found to result in a less accurate MCRDR engine than

decision trees. Comparisons were made between a decision tree SE and an induction rule SE. In each of these comparisons, the decision tree based SE built a more accurate KBS than the induction tree SE. A similar trend is reported by Compton et al (1995) who noted that C4.5 and Induct/RDR produced a more accurate KB than the normal induct method on its own. Figure 5-3 shows the decision tree and induction rules returned by See5 from the iris (UCI, 2012) dataset.

```
Decision tree:

petal_len <= 1.9: Iris-setosa (50)
petal_len > 1.9:
:...petal_wid > 1.7: Iris-virginica (46/1)
petal_wid <= 1.7:
:...petal_len <= 4.9: Iris-versicolor (48/1)
        petal_len > 4.9: Iris-virginica (6/2)

----------------------------------------------------

Rules:

Rule 1: (48/1, lift 2.9)
        petal_len > 1.9
        petal_len <= 4.9
        petal_wid <= 1.7
        ->  class Iris-versicolor  [0.960]

Rule 2: (50, lift 2.9)
        petal_len <= 1.9
        ->  class Iris-setosa  [0.981]

Rule 3: (46/1, lift 2.9)
        petal_wid > 1.7
        ->  class Iris-virginica  [0.958]

Rule 4: (46/2, lift 2.8)
        petal_len > 4.9
        ->  class Iris-virginica  [0.938]

Default class: Iris-versicolor
```

Figure 5-3. See5 decision tree (above dotted line) and induction rules from the iris dataset.

Table 5-2 below shows the MCRDR accuracies for two KBS built from a decision tree SE and an induction rules SE. The accuracy in this setup is simply the classifications the MCRDR engine got right, i.e. the number of correct classifications/total number of cases. The preliminary comparison of RDM and RM published in Maruatona, Vamplew, & Dazeley (2012) used induction rules simulated experts for all datasets, but subsequent tests, including the tests on which the results reported in chapter 6 are based on, used decision tree based simulated experts.

Dataset	Induction rules Accuracy	Dec. Tree Accuracy
Physical dataset	54%	81%
Car Evaluation	61%	88%

Table 5-2. MCRDR accuracy after learning from a See5 decision tree and induction rules SE.

5.6 Evaluation Metrics

Simple Accuracy/ Classifier Accuracy

The classifier accuracy (or simple accuracy) determines the system's ability to correctly classify a case. Dazeley (2007) calls this metric classification and defines it as the classifier's ability to correctly identify a case's class/group. The confusion matrix for simple accuracy uses two measures: True Classification (TC) and False Classification (FC), evaluated as follows:

- TC: Assigned to a case if the system correctly classified the case
- FC: Assigned to a case where the system failed to pick the right class

After the measures have been recorded, a system's simple accuracy is calculated as a proportion of the system's correct classifications (TC) on the whole dataset. The formula for simple accuracy is given by Equation (5-2):

$$Acc = TC/(TC + FC) \quad (5\text{-}2)$$

The ultimate objective of evaluating most systems is to determine how precise they are in predicting correct domain cases. This effectively means finding how effective a system is at picking correct classifications or giving correct answers. In a similar sense, comparing systems aims to find which has the highest accuracy over a number of domains. Learning systems such as RDR depend on the expert (simulated or human) for training.

In this project, the concept of Relative Accuracy (RA) is introduced whereby a system's RA is its accuracy as a proportion of the expert's accuracy. This impact of SE accuracy on a KBS accuracy has been previously used in RDR evaluations where a system is trained with simulated experts of various levels of expertise (Kang, Compton, & Preston, 1995). Given the

use of simulated experts of varying expertise, it is logical that the training system's ultimate accuracy will be influenced by its expert's competence level. This is affirmed by tests conducted by Kang, Compton & Preston (1995) using three levels of expertise. These tests showed that the clever expert produced a system with fewer errors than the stupid expert. In one dataset, the difference between the two systems error rates in testing was more than 10% (Kang, Compton, & Preston, 1995). For the tests in this thesis, RA measures the system's accuracy relative to the accuracy of the training SE. The formula for Relative Accuracy is defined like so:

$$RA = S_{Acc}/SE_{Acc} \quad (5\text{-}3)$$

where S_{Acc} is the system's accuracy (Acc) and SE_{Acc} is the SE's accuracy.

Prudence Accuracy

For prudence evaluations, the confusion matrix incorporates the warnings and whether they were issued at appropriate times. The following measures were used to evaluate individual RM and RDM predictions and are similar to the measures which the original inventors of RM and RDM (Dazeley, 2007; Prayote, 2007) used in their evaluations of the systems:

- FP: assigned to a case if the system produced a warning incorrectly.
- FN: assigned if a warning was required but the system failed to do so.
- TP: assigned to a case if a warning was produced correctly.
- TN: assigned to a case if the system did not produce a warning when it was not supposed to.

Simple accuracy has often been criticised for excluding class proportions and therefore not capturing the whole essence of the classifier's performance (Metz, 1978; García, Mollineda, & Sánchez, 2009). A typical illustration of this limitation is demonstrated in a faulty malware detection system where the system classifies all its 100 input files as benign by default. Given that six of the 100 files are actually malicious, the system would be rated as 94% accurate according to the simple accuracy metric, despite the defective classification component. In such a situation, a system could classify all cases as normal and potentially be given a high accuracy rating without actually detecting any frauds. This is potentially problematic in fraud detection especially where the data is known to be skewed with fraudulent cases comprising an average of less than 30% of the data (Phua, Lee, Smith, & Gayler, 2005; Phua, Alahakoon, & Lee, 2004). It has been suggested that accuracy should incorporate the classifier's rate of

positive and negative detections to avoid unbalanced performance ratings on skewed datasets (Metz, 1978). To this effect, two metrics; Specificity and Sensitivity are introduced.

Sensitivity or recall is the classifier's rate of detection of positive cases. Sensitivity has also been defined as the accuracy of positive cases or the proportion of correctly detected positive cases (Metz, 1978). For a prudence system, sensitivity can be viewed as measuring the accuracy of the system's warning mechanism since each TP corresponds to a correctly issued warning. Similarly, Specificity determines the classifier's rate of detection or the proportion of correctly detected negative cases. For prudence systems, Sensitivity corresponds to the instances when a warning was not necessary and the system correctly issued none. The formulas for Specificity and Sensitivity are given below:

$Se = \frac{TP}{P}$ and $Sp = \frac{TN}{N}$ (5-4)

Where P is the number of all actual positive cases and N is the number of all negative cases. Se and Sp can therefore also be defined thus:

$Se = \frac{TP}{TP+FN}$ and $Sp = \frac{TN}{TN+FP}$ (5-5)

From a prudence context, a positive case is one which is currently not covered by any rule in the knowledge base and hence necessitates the system to issue a warning. In a similar sense, a negative case would be one covered by one or more rules in the knowledge base and hence within the system's current knowledge. Ideally, a perfectly prudent RDR system will only issue warnings for cases beyond the system's current knowledge. In other words, warnings will only be issued if a rule for a particular case has not been added yet. This means that for a dataset with a perfect SE which has X rules, the system will only warn X times once before each of the X rules are added. After all X rules are added, there cannot be any more positives since the system's current knowledge is sufficient to cover all remaining cases. Observations done with a perfect SE (of four rules) on the Iris dataset showed that the MCRDR engine misclassified only four cases and each misclassification was immediately followed by a new rule addition. The same pattern occurred with a portion of the Car Evaluation dataset with 10 rules. Therefore the derivation of *P* and *N* for the Sensitivity and Specificity calculations could also be based on the idea that *P* (from equation 5-5) should be equal to the number of rules a dataset SE has.

Earlier it was shown that simple accuracy has limitations and that Specificity and Sensitivity measure opposite aspects of a system's accuracy. Prudence Accuracy or Balanced Accuracy

(BA) is a metric that incorporates the proportion of negative and positive classes in a dataset and avoids the problem of assigning classifiers exaggerated performance ratings on skewed datasets (Powers, Goldszmidt, & Cohen, 2005; Hardison, et al., 2008). BA is essentially a sum of the average accuracies of the positive (Sensitivity) and negative (Specificity) classes. Equation (5-6) defines the formula for BA.

$BA = 0.5(Se + Sp)$ (5-6)

BA avoids situations where the classifier is rated high because of an imbalanced dataset. For example, consider the earlier example where a faulty detection blindly classified 100 cases as benign (negative); using BA, the system's accuracy will be calculated as follows:

Se	= (0/6)	= 0, and
Sp	= (94/100)	= 94, so
BA	= 0.5(0+94)	= 47%

The example above shows how a system rated 94% according to simple accuracy can be more appropriately rated 47% if class imbalances are considered. There are a host of other performance metrics used in different domains to evaluate a range of classifiers (García, Mollineda, & Sánchez, 2009). The metrics described in this project are relevant and appropriate for the purposes of the reviewed systems; the ability to give correct classifications and issue warnings only when necessary.

Testing with Prudence

Prudent systems ideally issue warnings only when a warning is required. The primary measure of a prudent system's performance is prudence accuracy (specified in Equation 5-6), which involves measuring a prudence system involves determining if the correct classification was given, if a warning was issued and whether the warning was necessary.

For RDR specifically, prudence may also affect the classifier accuracy since rules are added only when a warning is issued. Consequently, this means that rules may not be added to the RDR knowledge-base due to missed warnings, potentially resulting in reduced classifier accuracy. This research therefore considers the impact of RDR prudence on classifier accuracy. This issue had been previously investigated by Dazeley & Kang (2008), who advised that the impact of missed warnings to the classifier accuracy in a prudent system was insignificant. It was further reported by Dazeley & Kang (2008) that significant impact of

missed warning on a prudent system's classifier accuracy was observed when the classifier already had a very low level of accuracy or in datasets where slight changes in attribute values resulted in a different conclusion altogether. This issue is explored further in this project with RDM and IPA.

The evaluation process used for each system in this thesis is as follows for each randomised dataset:

- An RDR classifier is built using the SE for the dataset without prudence. The classification accuracy is then measured using the process described in Algorithm 5-2. The classification accuracy serves as the base accuracy to be compared against the classification accuracy recorded when the system is run with prudence.
- Using the same SE, a second classifier is built with prudence enabled. The prudence accuracy and classifier accuracy of the system are measured using the process defined in Algorithm 5-4 below:

Algorithm 5-4

1. Accept a new case from randomised data.
2. Evaluate case against assessed KB.
3. Apply prudence
4. Evaluate case against SE.
5. If the case is correctly classified (If SE conclusion matches KBS):
   - If warning is issued, increment FP
   - Else increment TN
6. If the case is not correctly classified (If SE conclusion does not match KBS):
   - If warning is issued, increment TP and
     Add rule (or rules) to new KB to correspond to new case
   - Else increment FN
7. Go to step 1 for new case.

After a system's performance measures have been collected, the prudence accuracy can then be calculated according to the formulae in Equations 5-4, 5-5 and 5-6. The classifier accuracy with the prudence component turned on can be calculated as in Equation 5-2 where the rate of correct classifications is measured. Alternatively, this can be also

calculated from the pseudo-code above by adding the TNs and FPs. The two measures effectively represent a correct classification (TC).

Comparing the systems accuracies with and without prudence will also provide further insight on whether the system's missed warnings (FN) have compounding effects in terms of the overall accuracy of the prudence system, hence confirming or refuting whether the earlier tests conducted by Dazeley and Kang (2008) apply to the RDM and IPA prudent systems.

This evaluation process was repeated for each system for each of the datasets used in this project. The next section discusses the public datasets used to test and evaluate the systems.

5.7 Datasets

Eight different datasets have been used in testing and evaluating the three prudent RDR algorithms; RM, RDM and IPA. Each dataset is randomised before every test. Seven of these datasets included public datasets from the UCI Machine Learning repository (UCI, 2012). The public datasets included the Iris plants dataset, Car Evaluation, EMG Physical Action, Poker, Tic tac toe, Garvan and the Adult census income dataset. The other dataset is the proprietary online banking transactions provided by a bank. Further details on this dataset are discussed fully in chapter 7. A SE had to be developed for each of these datasets before tests could be run. The respective SE's were then used to train and test the systems using the approach described in the earlier sections. Table 5-3 describes each of the public datasets in terms of the dataset size, the number of decision tree rules from each dataset's SE and the SE's accuracy (for the dataset).

Dataset	Size/cases	Type	Attributes	Classes	DT size	SE Accuracy
Iris Plants	146	numerical	4	3	4	100%
Car Evaluation	1728	categorical	6	4	81	94%
Physical Action	1375	numerical	8	3	100	98%
Poker	5000	numerical	10	6	83	60%
Tic tac toe	958	categorical	9	2	172	100%
Adult	2000	categorical	8	2	94	85%
Garvan	5000	categorical	31	30	278	98%

Table 5-3. Description of used public datasets and SEs.

The DT size is the number of rules in the SE's decision tree. The SE Accuracy is the accuracy of the See5 decision tree simulated expert on a given dataset. The whole Garvan dataset has a total of around 21000 cases. Both RM and single class RDR had been tested on some part of this dataset (Dazeley, 2007; Prayote, 2007). In this project, no more than 5000 randomly selected cases were used for bigger datasets such as Garvan and Poker.

Keeping the dataset at a maximum of 5000 cases was both for simulation convenience and also within the limits of what a commercial RDR system was reported to handle. Labwizard, an RDR chemical pathology system was reported to have processed a maximum of around 340000 reports (cases) across 18 KB's in one month (around November 2005) (Compton, Peters, Edwards, & Lavers, 2005). This simplifies to at most 19000 reports per KB over a month, which is at least 633 cases per report per day for a 30 day month or 950 cases for a 20 day month. In another graph, it is shown that an individual KB did not exceed 140 rules per month (Compton, Peters, Edwards, & Lavers, 2005). The results in Table 5-4 are extracted from the Labwizard statistics, showing the total number of cases interpreted, total number of rules added, cases per day and the number of months in use.

KB	Total Cases	Total Rules	Months	Cases per day
A	1490767	1061	29	2570
B	1333598	1091	18	3704
E’	187848	9307	29	324

Table 5-4. Labwizard statistics for 3 knowledge bases (Compton, Peters, Edwards, & Lavers, 2005).

Note that Table 5-4 assumes a 20 day month as in Compton et al (2005).

5.8 Chapter Summary

The evaluation methodology chapter presented a framework of how individual RDR methods will be tested and compared. The brief survey of general evaluation approaches in KBS sought to demonstrate how there is a variety of evaluation schemes focusing on a range of different KBS aspects. However there is still no adopted standard evaluation criterion for KBS in both commercial and academic applications. The latter sections of the chapter focused on the evaluation approach used in RDR and the specific metrics applied in this dissertation in measuring the class accuracy and prudence accuracy of RM, RDM and IPA across a range of datasets. Class accuracy measures the system’s raw classification ability without prudence. Prudence accuracy incorporates the system’s warning mechanism and measures how effective the warnings are. The next chapter presents the results and analysis for all tests using the metrics discussed in this chapter on the public datasets.

6. Results on Public Datasets

6.1 Introduction

Using the metrics and methodology detailed in the previous chapter, this chapter presents a number of results and comparisons of prudence accuracies of RM and RDM on categorical and numerical public datasets. The chapter also presents results on the system's simple accuracies after their prudence mechanism are switched on and discusses if prudence has any effect on a system's simple accuracy. In the latter part of the chapter, IPA is compared with RM and RDM to verify if combining the two methods has any benefits in terms of improvement both in classifier and prudence accuracy.

6.2 Single and Multiple Classifications RDM

As explained in Chapter 4, the original Ripple Down Models (RDM) approach by Prayote (2007) used a Single Classification RDR engine. The modification to Multiple Classifications RDM (MC-RDM) proposed in this thesis and detailed in chapter 4 enables the system to handle both single classification and multiple classifications domains. Richards (2009) reported that even for single class domains, MCRDR produced a more compact KB than single class RDR.

Two configurations of Prayote (2007)'s reported SC-RDM results on the Garvan dataset (UCI, 2012) were compared with their MC-RDM versions on the same dataset to determine if Single Class RDM (SC-RDM) had any advantage or disadvantage over MC-RDM on a single class dataset. The statistics for SC-RDM presented on Table 6.1 are adapted from Prayote (2007)'s tests comparing different OECA thresholds on the Garvan dataset. The two systems ($SC\text{-}RDM_A$ and $SC\text{-}RDM_B$) are in fact the same SC-RDM method at two different configurations; one with a default OECA threshold of 0.0 (for $SC\text{-}RDM_A$) and one with 1E-10 (for $SC\text{-}RDM_B$) (Prayote, 2007). The MC-RDM systems are the multiple classifications equivalents of the RDM method proposed and introduced in this project. Table 6-1 presents two configurations of the SC-RDM and MCRDM's balanced accuracies on the Garvan dataset.

System	TP %	TN %	FP %	FN %	Se	Sp	BA %
SC-RDM$_A$	1.3	89.5	9.2	0.01	99.24	90.68	94.96
MC-RDM$_A$	7.8	84.7	4.4	3.1	71.56	95.06	83.31
SC-RDM$_B$	1.3	98.0	0.6	0.1	92.86	99.39	96.12
MC-RDM$_B$	7.2	90.8	1.4	0.6	92.31	98.48	95.39

Table 6-1. Comparison of two SC-RDM and MC-RDM configurations on prudence accuracy on the Garvan dataset

The two SC-RDM systems represent the most accurate (SC-RDM$_B$) and least accurate (SC-RDM$_A$) OECA threshold options reported by Prayote (2007). According to the results in Table 6-1, the prudence accuracy of both versions of the SC-RDM system is slightly higher than the accuracy of MC-RDM with corresponding thresholds. Generally, the two systems' accuracies are comparable given that the results include only one dataset. It must also be noted that the results for the MC-RDM systems were recorded from a random subset of the Garvan dataset whereas the reported results for the SC-RDM system were recorded from the whole Garvan dataset. Consequently, there is no guarantee that the simulated experts of the two systems will be similar both in terms of size and accuracy. The comparisons presented in Table 6-1 are therefore not definitive but give a good indication of the two systems' comparable performance.

Table 6-1 presents the only results on public datasets reported by Prayote (2007) on the SC-RDM system. Consequently further comparisons between SC-RDM and MC-RDM were impossible unless a SC-RDM system was developed. Given time constraints, this was unfeasible. The OECA threshold for the MC-RDM was set to a default of 0.0 for all other comparisons and evaluations. The next section evaluates the class accuracy of MCRDR across seven public datasets.

6.3 Simple Accuracy: MCRDR

Numeric datasets

Three public numerical datasets of up to 5000 cases and with SE decision tree sizes of up to 100 rules were tested on the MCRDR system. The MCRDR system represents both the RDM and RM systems without their prudence components. Results on Tables 6-2 to 7-4 represent averages over ten runs of each dataset with different randomisations of the data order. Table 6-2 presents results of MCRDR's simple accuracies in three numerical datasets.

Dataset	TC %	FC %	Acc %	RA %
Physical	59.4	40.6	59.4	60.6
Poker	52.0	48.0	52.0	86.7
Iris	97.3	2.7	97.3	97.3

Table 6-2. MCRDR's Simple accuracy in numerical datasets

MCRDR's average simple accuracy across numerical datasets is 69%. The system's relative accuracy on the numerical datasets is 81%. Essentially, this means that the system has correctly learnt 80% of the knowledge from the simulated expert. The next section compares MCDRD's class accuracies on four categorical datasets.

Categorical datasets

In the same manner as the evaluations with the numerical datasets, MCRDR's non-prudent classification accuracy was tested on four public categorical datasets. Table 6-3 presents the simple accuracy results for the system on the Car Evaluation, Tic tac toe, Garvan and the Adult datasets. A full description of all the datasets used in this project is given in the previous chapter.

Dataset	TC %	FC %	Acc %	RA %
Car	65.0	35.0	65.0	69.1
Tic tac toe	68.9	31.1	68.9	68.9
Garvan	89.0	19.0	89.0	90.8
Adult	62.5	37.5	62.5	73.5

Table 6-3. MCRDR's simple accuracy on categorical datasets

The results on Table 6-3 indicate that MCRDR's average simple accuracy on the categorical datasets is around 71%. In terms of relative accuracy, the system's average was 76%, indicating a fair learning ability. The next section summarises MCRDR's simple accuracies in both categorical and numerical datasets.

Analysis on RM and RDM's simple accuracy

Based on the simple accuracy experimental results presented in the previous sections, it is fairly evident that MCRDR recorded comparable results in both categorical and numerical datasets. The system's average simple accuracies in the numerical and categorical datasets were 69% and 71% respectively, indicating a consistent level of performance in both types of datasets. Another aspect of MCRDR's consistency is demonstrated in MCRDR's average RA in the two types of datasets, which are 81% for numeric data and 76% for categorical datasets. This suggests that MCRDR will on average learn at least seven tenths of the knowledge it gets trained on. The next section presents RM and RDM's prudence accuracies.

6.4 Prudence Accuracy: RM versus RDM

Numerical datasets

Table 6-4 presents the Specificity (*Sp*), Sensitivity (*Se*) and Prudence Accuracy (or BA) results of RM and RDM in the three numerical datasets. Three configurations of the RM system with three z step modifier values of 0.1, 0.5 and 0.95 for RM_A, RM_B and RM_C respectively were evaluated. A detailed analysis of the use of the z step modifier is given in chapter 4.

Balanced Accuracy (BA) is meant to address simple accuracy's limitation of ignoring class proportions by taking into account a dataset's negative and positive classes. This prudence accuracy metric considers whether a warning was issued by the system and whether the warning was necessary.

Dataset	System	TP %	TN %	FP %	FN %	Se %	Sp %	BA %
Physical	RDM	27.0	43.8	15.3	13.9	66.01	74.11	70.06
	RM_A	27.6	43.0	16.4	13.0	67.98	72.39	70.19
	RM_B	24.4	42.3	17.1	16.2	60.10	71.21	65.66
	RM_C	23.0	42.0	17.4	17.6	56.65	70.71	63.68
Poker	RDM	29.0	39.2	12.8	19.0	60.42	75.38	67.90
	RM_A	29.1	42.1	10.4	18.4	61.26	80.19	70.73
	RM_B	29.3	41.8	10.7	18.2	61.68	79.62	70.65
	RM_C	29.1	42.1	10.4	18.4	61.26	80.19	70.73
Iris	RDM	2.7	95.2	2.1	0.0	100.00	97.84	98.92
	RM_A	2.7	95.2	2.1	0.0	100.00	97.84	98.92
	RM_B	2.7	95.2	2.1	0.0	100.00	97.84	98.92
	RM_C	2.7	95.2	2.1	0.0	100.00	97.84	98.92

Table 6-4. Specificity, Sensitivity and prudence accuracy (BA) of RM and RDM in numerical datasets

In the Physical and Iris dataset, the two systems' prudence accuracies are comparable although an RM version (RM_A) is slightly ahead of RDM in the former dataset. In the Poker dataset, RM recorded a slightly higher BA performance than RDM.

In terms of average prudence accuracy on the numeric datasets, the results in Table 6-4 show a slight performance advantage of RM over RDM. In the numerical dataset, an RM version (RM_A) produced the highest average BA of 79% compared to RDM's average of 78%. An RM version has also recorded the highest BA in every numerical datasets. The two systems prudence performances in the numerical datasets are generally similar. There does

not seem to be an obvious trend in terms of which RM version is better with numerical data. The next section presents more BA results on categorical datasets.

Categorical datasets

Table 6-5 shows RM and RDM's Specificity, Sensitivity and prudence accuracy results over four categorical datasets.

Dataset	System	TP %	TN %	FP %	FN %	Se %	Sp %	BA %
Car	RDM	21.1	42.7	23.4	12.6	62.61	64.60	63.61
	RM_A	21.2	31.4	33.6	13.8	60.57	48.32	54.44
	RM_B	21.2	33.0	32	13.8	60.57	50.77	55.67
	RM_C	21.2	31.4	33.6	13.8	60.57	48.32	54.44
Tic tac toe	RDM	23.5	51.1	17.8	6.9	77.30	74.17	75.73
	RM_A	20.8	50.8	18.0	10.0	67.53	73.84	70.68
	RM_B	20.8	50.8	18.0	10.0	67.53	73.84	70.68
	RM_C	20.8	50.8	18.0	10.0	67.53	73.84	70.68
Garvan	RDM	7.8	84.7	4.4	3.1	71.56	95.06	83.31
	RM_A	7.8	81.0	8.1	3.1	71.56	90.91	81.23
	RM_B	7.5	81.0	8.1	3.4	68.81	90.91	79.86
	RM_C	7.6	81.1	8.0	3.3	69.72	91.02	80.37
Adult	RDM	29.4	45.7	18.8	6.1	82.82	70.85	76.83
	RM_A	31.0	41.8	20.7	6.5	82.67	66.88	74.77
	RM_B	31.1	42.2	20.3	6.4	82.93	67.52	75.23
	RM_C	31.0	41.8	20.7	6.5	82.67	66.88	74.77

Table 6-5. RM and RDM's Specificity, Sensitivity and BA in categorical datasets

According to the results, RDM had a better Sensitivity than RM in three datasets and RM produced better Sensitivity results than RDM in one of the four datasets. On average, RDM recorded a higher Sensitivity than RM, reaching 72% for the former and 70% for the latter. RDM's average Specificity in the categorical datasets is 74%, compared to RM's average of 70%.

RDM posted the highest BA in all four categorical datasets and had the highest average BA of 75%. The average BA of RM's three versions on the same datasets was 72%. This suggests that RDM may be slightly better than RM in categorical data although the relatively comparable average results of the two systems on categorical datasets indicate that the efficiencies their prudence methods are somewhat equivalent.

There does not appear to be a dominating version of RM across the categorical datasets despite RM_B recording the best prudence accuracy in two of the three datasets where the RM systems' BA is not the same for all three versions. Generally, there does not seem to be a best RM version in the categorical datasets.

Analysis on RM and RDM's prudence accuracy

RDM recorded a better prudence accuracy than the average RM versions in four of the seven tests. Conversely, RM recorded a higher prudence accuracy of the two systems across two of the numerical datasets. The two systems' prudence was equal in the Iris dataset. In general, RDM is ahead of RM in terms of the average BA across all datasets. This was mainly due to RDM's consistent advantage over RM on the categorical datasets. RDM's average BA across all datasets was 77% and RM's overall average BA was 76%. It must be mentioned also that the two systems' performances seem to be complementary in numerical and categorical datasets. The results show that RM appears to have a slight upper hand in the numerical datasets and that RDM was consistently the clear favourite in categorical datasets.

On average, there does not appear to be a clear best version of RM. The results presented in Tables 6-4 and 6-5 do not indicate a trend in terms of an outstanding RM version. It is worth noting that the differences between the prudence accuracies of the different RM versions per dataset are inconsistent, ranging from as low as 0.08% in one dataset to as high as 6.5% in another. In two of the seven datasets, the three RM systems recorded equivalent BA's. The inconsistent ordering of the best RM version over seven datasets suggests that the ideal setting and optimum version for RM seems to vary over different datasets.

The generally accepted norm with most data mining approaches is that each given method will usually have a specialisation domain, and may not be consistently superior to other methods in all domains- this is the so called 'no free lunch' theorem (Wolpert & Macready, 1997). The inconsistent performance of RM versions in different datasets and complementary domination of RM and RDM in different types of datasets support this proposition. The same sentiment is echoed in an anomaly detection context that a good algorithm/system should be suited to the user's dataset and domain and that a system may be selected for a specific domain although it may not be superior in other domains (Hodge & Austin, 2004) . The results at tables 6-4 and 6-5 suggest that RM might be better suited for numerical datasets and RDM for categorical data. The next section evaluates the systems simple accuracies when the prudence components are engaged and analyses whether prudence results in some form of improvement in simple accuracy.

6.5 Simple Accuracy Before and After Prudence

Table 6-6 shows RM and RDM's simple accuracies before the prudence mechanism is engaged and after. The simple accuracy results before prudence are as shown in Tables 6-2 and 6-3. The simple accuracy results after prudence were calculated by adding a system's False Positives (FP) and True Negatives (TN) which is equivalent to summing up the system's correct classifications (TC). When a system makes a correct classification (TC), the system's prudence mechanism can either issue a warning (FP) or not (TN). This is the rationale behind calculating a system's TC this way. The different RM versions given in the table correspond to the best RM version for a given dataset.

Dataset	System	Accuracy Before (%)	Accuracy After (%)	% Improvement
Physical	RDM	59.4	59.1	-0.3
	RM_B	59.4	59.4	0.0
Poker	RDM	52.0	52.0	0.0
	RM_A	52.0	52.5	0.0
Iris	RDM	97.3	97.3	0.0
	RM_C	97.3	97.3	0.0
Car	RDM	65.0	66.1	0.1
	RM_C	65.0	65.0	0.0
Tic tac toe	RDM	68.9	68.9	0.0
	RM_A	68.9	68.8	-0.1
Garvan	RDM	89.0	89.1	0.1
	RM_B	89.0	89.1	0.1
Adult	RDM	62.5	64.5	2.0
	RM_A	62.5	62.5	0.0

Table 6-6. Comparison of simple accuracy before and after prudence

Generally speaking, there does not appear to be an improvement or drop in classification accuracy after prudence. Either one of the two systems recorded an equal classification accuracy after prudence as the original MCRDR classification accuracy in six of the seven datasets.

Specifically, RDM's prudence produced a small improvement in simple accuracy in three categorical datasets and recorded small drop in simple accuracy in one numerical dataset. RDM's average classification accuracy improvement over the seven dataset is 0.27%. RM posted an insignificant accuracy improvement in one numeric dataset and recorded a small accuracy drop in one categorical dataset. RM's average classification accuracy improvement across the seven datasets is 0%. In other words, RM neither improved nor decreased MCRDR's base classification accuracy.

The two systems' average simple accuracy improvements are small and suggest that prudence has very little effect on the base classification accuracy. The results in Table 6-6 are unsurprising given that prudence primarily attempts to enable the system to issue some

kind of signal when a novel pattern or previously unseen case is introduced. The importance of rapid detection of novel cases has already been highlighted in chapter 3 as being critical in fraud detection. The results in Table 6-6 suggest that prudence provides this capability without compromising the classifier's simple accuracy. The results also agree with Dazeley & Kang (2008)'s earlier proposal that the effect of missed warnings in a prudent classifier are relatively minor.

6.6 IPA Prudence Accuracy

RM and RDM use different approaches in deciding whether to issue a warning for a case. RM is structural and depends on context derived from the MCRDR structure to train an ANN. RDM is attribute based and employs outlier detection methods on homogenised profiles. The two systems can be viewed as complementary, given their different development approaches and the different performances in accuracy and prudence. Part of this project's hypothesis was that merging a structural based prudence system (RM) with an attribute based prudence system (RDM) would result in some improvement in accuracy. The Integrated Prudence Analysis (IPA) method was developed from joining RM and RDM's secondary classifiers; RM's ANN and RDM's outlier detection methods (OEBA and OECA). The next sections present and analyse IPA's simple accuracy and balanced accuracy in the seven datasets used for the other two systems.

Three configurations of IPA were developed and tested. The three versions of IPA include IPA_{OR}, IPA_{AND} and IPA_{ANN}. IPA_{OR} and IPA_{AND} configurations involve the joining of RM's ANN output and RDM's aggregated outlier index through an AND or OR connection. In the IPA_{ANN} version, MCRDR rule paths are added to the aggregated outlier index from MC-RDM's outlier detectors and all fed to the ANN. A detailed description of IPA is given in chapter 4. Since IPA uses a similar MCRDR engine as the RDM and RM systems, it's simple accuracy results were similar to the two systems' simple accuracies shown in Tables 6-2 and 6-3.

Numerical datasets

Table 6-7 presents the prudence accuracies of different IPA versions on numerical datasets.

Dataset	System	TP %	TN %	FP %	FN %	Se %	Sp %	BA %
Physical	IPA_{AND}	27.6	41.7	17.7	13.0	67.98	70.20	69.09
	IPA_{OR}	29.7	39.9	19.5	10.9	73.15	67.17	70.16
	IPA_{ANN}	26.5	40.7	18.7	14.1	65.27	68.52	66.89
Poker	IPA_{AND}	24.9	45.4	7.1	22.6	52.42	86.48	69.45
	IPA_{OR}	29.1	45.2	7.3	18.4	61.26	86.10	73.68
	IPA_{ANN}	26.1	45.2	7.3	21.4	54.95	86.10	70.52
Iris	IPA_{AND}	2.7	95.2	2.1	0.0	100	97.9	98.9
	IPA_{OR}	2.7	95.2	2.1	0.0	100	97.9	98.9
	IPA_{ANN}	2.7	95.2	2.1	0.0	100	97.9	98.9

Table 6-7. IPA prudence accuracy on numerical datasets

According to the results in Table 6-7, IPA_{OR} has outperformed the other two versions in BA in two of the three numerical datasets and is close to the other two versions in the other dataset. IPA_{OR}'s domination over other IPA configurations in balanced accuracy in the numerical datasets is supported by the system's average BA of 80.9% compared to IPA_{ANN}'s 78% and IPA_{AND}'s 78.4%. BA results on categorical datasets are shown in the next section.

Categorical datasets

The three IPA versions' prudence accuracy results in the categorical datasets are presented in Table 6-8.

Dataset	System	TP %	TN %	FP %	FN %	Se %	Sp %	BA %
Car	IPA_{AND}	20.4	39.1	26.6	13.9	59.5	59.5	59.49
	IPA_{OR}	22.5	37.3	28.4	11.9	65.4	56.8	61.09
	IPA_{ANN}	21.5	38.7	27	12.8	62.9	58.9	60.79
Tic tac toe	IPA_{AND}	20.8	53.2	15.7	10.3	66.9	77.2	72.05
	IPA_{OR}	24.8	51.2	17.7	6.3	79.7	74.3	77.03
	IPA_{ANN}	22.5	51.8	17.1	8.6	72.4	75.2	73.76
Garvan	IPA_{AND}	7.5	85.2	3.9	3.4	68.8	95.6	82.22
	IPA_{OR}	8.1	84.3	4.8	2.8	74.3	94.6	84.46
	IPA_{ANN}	7.6	84.6	4.5	3.3	69.7	95.0	82.34
Adult	IPA_{AND}	28.5	44	18.5	9.1	75.8	70.4	73.10
	IPA_{OR}	32.8	43.4	19.1	4.7	87.5	69.4	78.45
	IPA_{ANN}	29.1	43.7	18.8	8.4	77.6	69.9	73.76

Table 6-8. IPA prudence accuracy statistics on the categorical datasets

In categorical datasets, IPA_{OR} has continued to record the best prudence accuracy of the three IPA configurations. The pattern observed in the numerical datasets has recurred in the categorical data, with IPA_{OR} producing the highest BA, followed by IPA_{ANN} and finally IPA_{AND}. The domination of IPA_{OR} over other IPA versions in six of the seven datasets suggests that IPA_{OR} is the best of the three versions in prudence accuracy. At this point, it is clear that IPA_{OR} is by far the best IPA version. The consistently higher BA ratings in both the categorical and numerical datasets support this proposition.

Although the IPA versions used in these tests include IPA_{ANN}, IPA_{OR} and IPA_{AND}; these are not the only possible configurations of IPA. Many other ways of combining RM and RDM exist and some could possibly have better accuracies than any of the versions involved in these

tests. However, given the current results, IPA_{OR} has emerged as the most optimum tested configuration of IPA.

6.7 Integrating Two Prudent Methods: Does It Work?

The original intent behind IPA was to use RM and RDM to complement each other such that the combination is better than using any of the two methods individually. The combination of attribute based RDM and structural based RM was anticipated to eliminate the individual methods' limitations, consequently improving accuracy. The next two tables present the accuracies of IPA_{OR} (the best IPA version), RM and RDM in numerical and categorical data.

Numerical datasets

For RM, the results shown on Table 6-9 were taken from the best RM configuration in each dataset. Table 6-9 shows the three systems' prudence accuracy in numerical datasets.

Dataset	System	BA %
Physical	RDM	70.06
	RM_A	70.19
	IPA	70.16
Poker	RDM	67.90
	RM_C	70.73
	IPA	73.68
Iris	RDM	97.3
	RM_A	97.3
	IPA	97.3

Table 6-9. IPA, RM and RDM's BA in numerical datasets.

IPA recorded the best prudence accuracy in two of the three datasets. The integrated prudence method had the best average BA of 79.6% on numeric datasets, followed by RM at 78.6% and RDM at 78.3%. The average prudence accuracies of the three systems on the

numeric datasets are relatively close to one another, although IPA appears to have slight advantage over the other two systems. Results on the three systems' prudence accuracy on categorical datasets will confirm whether this is true for all types of data or only for numerical datasets.

Categorical datasets

Table 6-10 presents RM, RDM and IPA's balanced accuracy in categorical data.

Dataset	System	BA %
Car	RDM	63.61
	RM_B	55.67
	IPA	61.09
Tic tac toe	RDM	75.73
	RM_A	70.68
	IPA	77.03
Garvan	RDM	83.31
	RM_C	80.37
	IPA	84.46
Adult	RDM	76.83
	RM_B	75.23
	IPA	78.45

Table 6-10. RM, RDM and IPA Acc and BA on categorical data

In categorical data, IPA produced the best prudence accuracy in three of the four datasets. Once more the combined system had the highest average prudence accuracy in the categorical datasets. IPA recorded the best average BA of 75.3%, with RDM and RM following at 74.9% and 70.5% respectively.

The results in Tables 6-9 and 6-10 suggest that IPA has some advantage over RM and RDM in terms of prudence accuracy. Although the performance differences between IPA and the two other systems are relatively small (between 0.3% and 3%) in each dataset, IPA's consistent recording of the highest BA in five of the seven datasets suggests that the combined method is better than any of the individual method in most instances.

It is also worth noting the strength of selected RM versions in particular datasets. This shows that at the right settings, RM is as competitive as the RDM and IPA. However pre-determining this optimum setting (the step modifier) is still not obvious as the results differ in different datasets.

IPA Simple Accuracy after Prudence

As was done with RM and RDM, the simple accuracy of IPA was compared to the base system (without prudence) to determine if prudence had an effect on the system's simple accuracy. Table 6-11 records the MCRDR base system's simple accuracy against IPA's simple accuracy.

Dataset	Accuracy Before (%)	Accuracy After (%)	% Improvement
Physical	59.4	59.4	0.0
Poker	52.0	52.5	0.5
Iris	97.3	97.3	0.0
Car	65.0	65.7	0.7
Tic tac toe	68.9	68.9	0.0
Garvan	89.0	89.1	0.1
Adult	62.5	62.5	0.0

Table 6-11. IPA's simple accuracy before and after prudence

According to the results in Table 6-11, IPA has recorded minor classifier accuracy improvements in three of the seven datasets. On four occasions, IPA maintained an equal simple accuracy as the base MCRDR system. IPA's average simple accuracy improvement across the seven datasets is 0.2%. Once more, the results suggest that prudence has a very little effect on MCRDR classifier accuracy.

Importantly, the IPA results in Tables 6-9 to 6-11 confirm this project's hypothesis that combining the two complementary prudence methods in some fashion could improve the

individual systems' prudence accuracy. IPA has shown a consistent performance advantage over RM and RDM according to the results presented in Tables 6-9 to 6-10.

6.8 Chapter Summary

This chapter presented a number of evaluations, comparisons and analysis between different systems on public datasets. The early sections presented results and analysis of class accuracy and prudence accuracy comparisons between the RM and RDM systems in numerical and categorical public datasets. The simple accuracy results of RM and RDM were also taken after the systems' prudence components were switched on. Results show that prudence as applied in this project has very little effect on a classifier's simple accuracy. The latter sections showed prudence accuracy results from three IPA versions and selected the best IPA configuration which was then evaluated against the two methods it was built from (RM and IPA). Although only three configurations of IPA were developed; IPA_{ANN}, IPA_{OR} and IPA_{AND} do not represent the only possible combination options between RM and RDM.

Results from throughout the chapter suggest that RDM generally has a higher BA than RM in categorical datasets and that a combination of RM configurations has a higher prudence accuracy than RDM in numerical datasets. It was also shown in the latter sections of the chapter that combining the two systems (RM and RDM) into IPA does indeed improve prudence accuracy. The next chapter presents test results on IPA with Internet banking transactions and discusses the potential use of IPA in a commercial online banking fraud detection system.

7. IPA Results on Internet Banking Data

7.1 Introduction

Chapter 7 presents results from IPA's evaluations on Internet banking transactions. The Internet banking data was sourced from a commercial online banking fraud detection system and has been depersonalised before it was used on this project. The chapter describes the banking data, the tests done with IPA and the results in terms of balanced accuracy. A simple comparison between IPA and a commercial Internet banking fraud detection system is also given, using the commercial system's performance metrics. The latter sections of the chapter provide some remarks on the effective use of research in developing useful solutions in Internet banking fraud detection. A few recommendations on successful online banking fraud detection are also briefly discussed including a hierarchical approach to screening online transactions and on-going user education.

7.2 Internet Banking Fraud

Financial institutions and individuals continue to lose millions of dollars through Internet banking fraud. The loss of money through online banking is mainly perpetrated by some form of identity theft. After a legitimate user's online banking credentials have been stolen somehow, their account is accessed illegally and their funds transferred to a mule's account. The mule then withdraws the stolen cash from their account and wires it to the ultimate fraudster. This is usually how Internet banking accounts are robbed and the money transferred by mules to cyber-criminals who exploit international policing loopholes. Additionally, the banks have to decide whether to launch costly, slow investigations or reimburse the lost funds.

The main thing to note in Internet banking fraud is that access to a victim's online banking account was achieved using stolen but correct credentials. These details are usually sought from unsuspecting victims through sophisticated phishing and pharming techniques. A July 2011 RSA Fraud Report noted that Internet users were being persuaded, encouraged, conned, swindled and reasoned into revealing their passwords and important credentials and also clicking malicious links that downloaded malware into their computers (RSA, 2011). According to this report, the sophisticated social engineering operations use the same

persuasion techniques employed by modern corporate advertising and marketing. In other cases, the fraudsters developed complex pharming Trojans that modified a victim's computer's hosts file so that an Internet banking site's IP address is redirected to a phishing website where online banking credentials are illicitly obtained and sent to the fraudster (RSA, 2011).

In response to the advancing pharming and phishing campaigns, the focus in deterring online banking frauds must not be exclusively focused on preventing entry (password checks, dynamic screen keyboards, etc) but also on detecting fraudulent activity within an account. The increasing deployment of advanced and sophisticated phishing techniques by fraudsters to get users' details suggests that Internet banking systems must have capabilities to spot illegitimate access even if the password and username are correct. This is the fundamental approach adopted by this research, strengthening fraud detection mechanisms within accounts and not just before access is granted.

A brief review of some commercial Internet banking fraud detection systems was given in chapter 2. Usually, not much is publicised about how these systems work for competition and security reasons. The reviewed systems (PRM, Falcon and SAS), which are the most popular in the industry have a relatively common architecture. The systems use either anomaly detection or a signature detection approach or a combination of the two to define Internet banking users' spending, access, cash withdrawal, transfer, bill payment and other transaction patterns. A rule-base or ANN system is then used to monitor, update and detect any variations on these patterns.

The approach proposed and developed by this project is similar in structure to the commercial systems but uses a new method and focuses on detecting fraudulent activity within a compromised account. The IPA system mainly combines two prudent RDR systems and is intended to be especially helpful in warning system administrators of new internet banking transaction patterns. Tests and evaluations with public datasets showed that a combination of RM and RDM was more accurate than any of the two systems individually. A full description of IPA and the evaluations with the public datasets is given in chapters 4 and 6 respectively. The next section describes the online banking transactions used to evaluate and establish IPA's worth in online banking fraud detection.

7.3 Internet Banking Transactions

Internet banking systems record a plethora of details every time a user accesses their account. Usually, a combination of these details are also used by fraud detection systems to distinguish a fraudulent transaction from a legitimate one e.g. an attempt to transfer more than the set limit, several, consecutive transfers to a new account in a short period of time or a relatively odd log-in time. These actions might well be by the legitimate account holder but if they are anomalies or variations of the user's usual patterns, the system combines them with other details to conclusively decide if the transaction is legitimate or not. For this project, Internet banking transactions from a commercial online banking system were used to test IPA.

Table 7-1 shows some of the attributes of an online banking transaction dataset used to test and evaluate the IPA system. Different transactions have respective names depending on the details of the transaction. For example, OTT represents Outward Telegraphic Transfers which are electronic international money transfers. BPay is a system through which utility bills and other service providers can be directly paid from the client's account. Funds Transfer (FT) and Pay Anyone (PA) are other types of an Internet banking transaction representing the movement of funds between accounts linked to the same holder and the transfer of funds to an account held by a different account holder respectively. Different banks and systems adopt different nomenclatures for different transactions.

Name	Description	Type	Values
Transaction ID	Unique ID for every transaction	label	System generated
Transaction Type	Type of transaction	discrete	BPAY, OTT, FT, PA
Account From	Source account number	label	System generated
Account To	Destination account number	label	System generated
Account Type	Type of account in use	discrete	Savings, Business, Credit, Home loan
Event time	Time of transaction	Time	System time
Session ID	Unique session ID	label	System generated
Browser String	String describing browser user	label	System generated
IP Address	IP address for machine in use	label	System generated
Country	Host country for given IP	Label	System generated
Trans Amount	Transfer amount (if Transfer)	Continuous	0-account balance
BPay Amount	BPay Amout (if BPay)	Continuous	0-account balance
IMT Amount	International transfer amount	IMT Amount	IMT Amount
Biller Code	Unique biller code (for BPay)	Label	System generated
Biller Name	BPay Biller business name	Label	System generated
Log in ID	User's log in ID	Label	System generated
Log in Time	Time of log in	Time	System time
Log in Count	Number of log ins for the day	Continuous	Real numbers
Password change	Number of password changes	Continuous	Real numbers

Table 7-1. Description of online banking transaction attributes

7.4 Obfuscation and Online Banking Data

Before the data was issued by the bank, intense obfuscation was carried out on personal details and thorough checks were conducted to ensure that no personally identifying details remained in the data whilst the data still retained its referential integrity. An Obfuscation tool (or Obfuscator) was developed to scramble personal details in a dataset such that the data could still be useful for research but contain no real personal details. The obfuscation process had to be irreversible and as such no algorithm should have the capacity to recover the scrambled personal details. Additionally, the internal relations of the different data elements should be maintained. This means that identical items had to remain identical after obfuscation.

The Obfuscator input file was a transaction log-file from a bank database similar to the sample shown in Table 7-1. The file may contain any number of headers and a sufficiently large number of items under each header. The Obfuscator uses this log-file (csv format) and produces an equivalent .csv file with the same headers as the input file but different data items according to what columns are to be scrambled. Any number of headers (or columns) can be selected to be obfuscated. Furthermore, the tool provide a choice of three obfuscation styles including numeric only obfuscation, alphabetic obfuscation and mixed (alpha-numeric) obfuscation. When the user preferences have been specified, obfuscation is done and the output file will have the same headers as the input, with the selected columns scrambled accordingly. By the time of use by this project, some features such as IP address, BPay Biller codes, session IDs and transaction IDs were de-personalized but maintained referential usefulness. Table 7-2 demonstrates the three types of obfuscation using some examples.

Obfuscation type	Original string	Obfuscated string
Alphabetic	John Columb	hnvr hacdekj
Alpha-numeric	John Columb	7n4r 5acde8j
Numeric	121.220.74.208	48.127.331.3105

Table 7-2. Demonstration of three obfuscation types

Data preparation

In total, the dataset comprised 40 attributes and consisted of 1760 transactions, 60% of which were legitimate and 40% of which were fraudulent transactions. The data was divided into three streams of exclusively categorical attributes (trans_Cat) consisting of eight attributes, exclusively numerical attributes (trans_Num) with seven attributes and the mixed attributes (trans_Mixed) comprising both categorical and numerical datasets. The division of transactions into categorical and numerical streams was to enable focused tests on the OEBA and OECA components of outlier detection in the RDM part of IPA. Simulated experts for the three streams (trans_Cat, trans_Num and trans_Mixed) were developed using See5. The SE for the trans_Cat dataset had 49 rules, the trans_Num SE had 72 rules and the trans_Mixed SE had 100 rules. The SEs had respective accuracies of 75%, 60% and 90%.

IPA performance in Online Banking Fraud Detection

Table 7-3 shows IPA's cross validated accuracy results on the trans_Cat, trans_Num and trans_Mixed datasets.

Dataset	TC %	FC %	Acc %	RA %
trans_Cat	70.3	29.7	70.3	93.7
Trans_Num	53.3	46.7	53.3	88.8
Trans_Mixed	73.9	26.1	73.9	82.1

Table 7-3. IPA simple accuracy on online banking data

The main thing to note from Table 7-3 is that IPA's good learning capability is applicable to Internet banking data. This is shown by the system achieving an average of over 80% across the Trans_Cat and Trans_Num datasets both in simple accuracy and balanced accuracy. Importantly, IPA has shown a capability to learn over 80% of the knowledge from a simulated expert in the complete dataset (Trans_Mixed) with both categorical and numerical attributes. Table 7-4 presents IPA results on prudence accuracy.

Dataset	TP %	TN %	FP %	FN %	Se %	Sp %	BA %
trans_Cat	18.6	57.4	12.9	11.1	62.63	81.65	72.14
Trans_Num	27.4	43.3	10.2	19.1	58.92	80.93	69.93
Trans_Mixed	15.6	69.4	4.5	10.5	59.77	93.91	76.84

Table 7-4. IPA prudence accuracy on online banking data

According to the results in Tables 7-3 and 7-4 above, using exclusively categorical attributes produces a higher accuracy than using numerical attributes only. This is shown by the higher relative accuracy from the categorical datasets. This also suggests that OECA has a higher accuracy than OEBA in online banking transactions. It is also important to note that the system's accuracy in the mixed dataset is also very good.

Prudence in Fraud detection

The idea of prudent fraud detection systems in any domain is meant to enable the system with an added capability of reporting strange, suspicious or previously unseen behaviour. As much as simple accuracy is important in such systems, it is also important that the systems realise when a case is beyond their current expertise and be able to issue alerts or warnings rather than attempt to classify the case using the current knowledge. According to the results above, IPA has shown an overall adequate prudence across both datasets but especially in the categorical transaction details where the system learned more than 90% from the SE.

The importance of a capability like prudence in Internet banking fraud detection systems can never be understated. In 2008, two independent reports by IBM and SAS advised that the major challenge in online payment systems is the ability to react rapidly to new fraud trends (SAS, 2007; IBM, 2008). The sooner a peculiar case is reported to an administrator by the system, the better it is for the organisation. If strange cases are reported as soon as they are discovered by the system, new rules will be promptly developed and the system will be better equipped to handle the recurrence of such cases. However if the system attempts to classify a new case and misclassifies it, it may be both time consuming and inconvenient for the organisation to add a rule that rectifies the misclassified case. In some cases, the organisation may have to reimburse the defrauded account and investigate the case a few

days later and this would be much messier than if the system had reported the strange case the instance it was processed.

A potentially problematic scenario with prudence is when a system issues too many unnecessary warnings. In an Internet banking application for example, too many unnecessary warnings may overwhelm the human operators as they will have to check each reported case. Ultimately, the imperfection of systems and a need for some kind of warning capability in such systems results in a convenient compromise between warning too few times and warning too many times. In online banking fraud detection, it may be wiser for the system to warn too many times and detect most frauds than warn too few times and miss most frauds. Fortunately, the exact setting of the warning threshold in IPA is customizable and can be specifically set based on the organisational needs. Generally speaking, the good BA results recorded for both the numerical and categorical attributes suggest that IPA has good potential as far as application in online banking fraud detection is concerned.

7.5 IPA versus Commercial System

To get the relative significance of IPA's performance in online banking fraud detection, in-house statistics from a proprietary commercial system were compared to IPA's accuracy. The commercial system's performance metrics are different to the metric used previously to evaluate IPA but nonetheless, IPA's corresponding results on the commercial metrics could be calculated. The commercial system's metrics DR(n) and DR($) measure the system's rate of detection in terms of how many frauds were detected by the system as a percentage of all frauds (n) and in terms of how much in dollar terms the detected frauds represent over the whole fraud amount. The formula for Detection Rate (DR) is defined in equation (7.1).

$$DR(n) = \frac{Detected_{fraud}}{Total_{fraud}} \quad \text{and} \quad DR(\$) = \frac{Detected_{amt}}{Total_{amt}} \qquad (7.1)$$

Detection Rate (n) is the number of system detected fraudulent transactions over all fraudulent transactions including reported cases missed by the system. Detection Rate ($) refers to the amount of money represented by the detected fraudulent transactions as a percentage of the total amount of all fraudulent transactions in dollar terms. For example assuming that the system detected 10 of 20 fraudulent transactions and that the fraudulent transactions were directly responsible for a loss of $1000 to the bank, of which $375 was

recovered when the system detected the fraud; DR(n) and DR($) can then be determined thus:

DR(n) = 10/20 = 50%

DR($) = 375/1000 = 37.5%

Detection rate as defined in equation (7.1) is equivalent to recall or sensitivity in classification systems. In chapter 5, specificity was defined as the proportion of correctly detected positive cases (Metz, 1978). For a prudent system such as IPA, this corresponds to the rate at which the system issued correct warnings or knew when a case needed further examination by the expert. This essentially means that sensitivity in IPA corresponds to cases when the system knew it would be potentially wrong because it had inadequate knowledge to give a decisive conclusion. Results on Table 7-4 show that IPA has a sensitivity of 60% on the mixed online banking transaction attributes. Table 7-5 below shows a comparison of IPA and a commercial system's detection rates.

System	DR (n)
commercial	65%
IPA (trans_Mixed)	60%

Table 7-5. IPA vs. Commercial system detection rates

The performance statistics for the commercial fraud detection system were sourced from an Australian bank. The IPA results are derived from the system's sensitivity, calculated from the results at Table 7-4.

A DR ($) rate could not be determined for either system because there was no indication of how much in dollar terms, the fraudulent transactions represented. The bank however advised that the total amount of money lost through fraud differs on a daily basis and that it would be ideal if the system could spot potential frauds as the transaction is conducted or before the funds are withdrawn or moved offshore.

It may not be fair to decide which of the two systems is better because the reported results are from a single batch of legitimate and fraudulent transactions compiled over time. The results may be different on a day to day setting. The bank advised that Internet banking transactions are usually skewed with legitimate cases comprising well over 70% of all

transactions, and even more than 98% according to Wei, Li, & Cao (2012) and Krivko (2010) . This may also affect a detection system's performance especially if there were adequate cases of both positive and negative cases to train the system. Generally, it is encouraging to note that IPA's detection rate is at least comparable to a commercial system.

7.6 Internet Banking Fraud Detection Framework

Although widely reported in the media and by some organisations as a problem, it is hard to measure how big a problem Internet banking fraud really is. This is because banks and other financial services providers are reluctant to disclose specific facts and figures about Internet banking fraud and the extent by which it affects their businesses (Kou Y. , Lu, Sirwongwattana, & Huang, 2004; Bolton & Hand, 2002). Consequently, it is harder for researchers to develop practical and immediately useful methods and systems for fraud detection in this domain. This is because real data is often hard to acquire and proprietary data cannot be shared between researchers in the same field (Stolfo S. , Fan, Lee, & Prodromidis, 1997; Bolton & Hand, 2002). The whole situation makes it especially difficult to develop targeted and impactful solutions through research. This is a serious deterrent to the effective use of research resources in solving real problems and has to be addressed accordingly.

Successful online banking frauds occur because of a number of security breaches including theft of Internet banking credentials which may have resulted from a phishing or pharming campaign. Essentially, a fraudster first has to get a user's login details before they can access their Internet banking account. Getting these credentials usually involves some other scam such as phishing or pharming. Conversely, curbing online banking frauds should involve a number of measures in other aspects of identity security, not just in a financial service provider's Internet banking login page. By the same token, Internet banking fraud cannot be realistically stopped by the use of a single software application. A hierarchical approach involving a number of specific screening and checking mechanisms in verifying and authenticating users' identities may help deter and detect fraud attempts. This approach, also known as layered security involves using different screening mechanisms at different stages of a transaction so that the different controls compensate and complement one another (EMA, 2012). A 2012 white paper by the Federal Financial Institutions Examination Council's (FFIEC) advises that layered security can substantially strengthen the overall

security of online services reducing account break-ins, protecting against loss of customer details and effectively preventing loss of funds (FFIEC, 2012).

Internet banking services user awareness and education is also critical in combating identity theft. Users' education on identity theft, the ways it can be perpetrated and safety measures should be continual in ensuring that they are informed of potential ways through which their identity can be stolen and ultimately result in distress, loss of money and other inconveniences. The RSA report (RSA, 2011) mentioned earlier has highlighted the fact that fraudsters use sophisticated methods of persuasion to lure unsuspecting victims into their scams. In this regard, users will need to be cautious of potential phishing and pharming scams and always exercise discretion before giving away their details.

IPA's impressive performance with Internet banking data presents a viable solution to online banking fraud detection. This is especially so given that most Internet banking frauds go undetected (or detected after the incident) because fraudsters use correct log in details and that most online banking security features focus on preventing entry into accounts than they do in detecting fraudulent activity within compromised accounts. IPA is versatile and can incorporate other data sources within the bank's systems to form a smarter Internet banking intelligence system. This system can then be used with other security mechanisms in a layered approach as suggested in the previous sections. The inherent advantage of an IPA based Internet banking intelligence system will be its ability to detect novel cases rapidly and warn a human expert (or system administrator) in time. Given its advantages over conventional knowledge-based systems, its accuracy and fast, cost effective knowledge acquisition methods, IPA is definitely worth a try in Internet banking fraud detection.

7.7 Chapter Summary

This chapter presented IPA's accuracy results on Internet banking transactions. Results from a commercial system used to detect frauds in online banking transactions were also given and benchmarked against IPA. Although the size of the data given to this project and other factors restrict some aspects of the evaluations, it is fair to conclude that given what has been observed (and recorded), IPA has shown good potential in online banking fraud detection. It is also the view of this project that effective fraud detection in Internet banking transactions cannot be realistically achieved from one individual system. This view is based on facts about how most frauds exploit a gap in users' ignorance, ineffectiveness and lack of

systems. The next chapter summarises the whole project and concludes the dissertation. The chapter will provide detail on what the project aim was, what was involved and done to achieve this aim and whether this aim was achieved.

8. Conclusion

8.1 Introduction

This chapter summarises the whole project and briefly describes current trends and statistics in Internet fraud especially phishing through which people's passwords get stolen and their online banking accounts robbed. A brief summary of present approaches to detecting Internet banking frauds is given, and the limitations of these approaches are identified. The chapter also highlights a possible solution to these limitations and proposes how these methods were used in the project. A brief analysis of the results, analysis and finally the issues encountered and future work are presented at the end of the chapter.

8.2 The Situation

Despite the advances and conveniences of buying and selling goods and services online, Internet crime or cybercrime poses a major problem for most businesses and individuals. One of the most prominent examples of internet fraud is Phishing, whose objective is to get hold of unsuspecting internet users' credentials and ultimately use these details to defraud people's accounts (RSA, 2010). According to the Anti-Phishing Work Group (APWG), the financial services sector remained the most targeted industry by cyber-criminals as at 2011 (APWG, 2011). Furthermore, the effectiveness (and success rate) of phishing was forecasted to rise due to the increase in the development of more sophisticated techniques by online fraudsters (RSA, 2010).

The ultimate motive for stealing people's online usernames and passwords is to log into the victims' accounts (undetected) and exploit their accounts by transferring funds to a different bank account. Most Internet banking sites unfortunately have poor detection mechanisms if the log-in credentials were correct. A common approach to detecting fraudulent activity in online banking systems involves the use of a conventional Knowledge-Based System (KBS) and an Artificial Neural Network (ANN). Three Internet banking fraud detection systems used in more than 40 countries, employed at 43000 Internet banking sites and used by eight of the world's top 20 banks have this architecture of combining a KBS and an ANN (FICO, 2011), (ACI Worldwide, 2011; SAS, 2007).Other systems employing Rule-Bases and ANN's in fraud detection include (Kou Y. , Lu, Sirwongwattana, & Huang, 2004; Phua, Lee, Smith, & Gayler,

2005; Weatherford, 2002). The worth and importance of KBS in a host of Industries and domains is unquestionable. KBS can imitate and learn human expertise, are faster, more consistent and cheaper to run than humans (Li, Xie, & Xu, 2011; Abraham, 2005).

KBS have also been criticised especially on their Knowledge Acquisition (KA) and Maintenance approach. The process of transferring knowledge to a KBS has been condemned for being slow and indirect and labour intensive (Richards, 2009; Dazeley, 2007; Giarratano & Riley, 2005). Maintenance in such systems has also come under scrutiny for being an additional and separate task from KA and ultimately lengthy and costly (Richards, 2009; Hayes-Roth & Jacobstein, 1994). Another limitation of KBS is their lack of awareness of their limitations- a phenomenon known as brittleness. Brittleness occurs when a KBS does not realise when its knowledge is inadequate for a particular case (Compton, Preston, Edwards, & Kang, 1996) and this leads to the system giving erroneous and sometimes illogical conclusions.

Motivated by the elimination of maintenance and KA limitations of KBS, the Ripple Down Rules approach (RDR) was founded around 1988. RDR eliminates the need for a knowledge engineer as the expert directly interacts with the system (Kang, Compton, & Preston, 1995). Additionally, maintenance and KA in RDR are integrated, usually trivial and brief (Kang, Compton, & Preston, 1995; Richards, 2009). There are two varieties of RDR; Single Class RDR (SC-RDR) and Multiple Classifications RDR (MCRDR). SC-RDR (usually simply called RDR) is binary in structure and can only produce a single conclusion from every case. MCRDR is an n-ary version of RDR and can give more than one classifications/conclusion to a case. Since its inception, RDR has been used in different applications including web browsers, help desk systems, online shopping and email management systems (Richards, 2009).

To address brittleness, the idea of Prudence was introduced to equip the KBS with the ability to issue warnings whenever a case was beyond the system's current set of expertise (Kang, Compton, & Preston, 1995). A perfectly prudent RDR system is therefore for all intents and purposes free of traditional KBS' maintenance, KA and brittleness limitations. In terms of fraud detection, a prudent RDR system should have a competitive edge over traditional approaches to fraud detection in that new fraud patterns will be detected rapidly through the system's warning mechanism. The two main existing prudent RDR techniques are Rated MCRDR (RM) and Ripple Down Models (RDM) (Dazeley, 2007; Prayote & Compton, 2006).

8.3 The Project

Given the demonstrated superiority of the prudent RDR methods over conventional KBS in maintenance and KA, this project sought to conduct a comprehensive evaluation and comparison of two of the best prudent methods: RM and RDM. There are currently no published records of such comparisons so the proposed evaluations should provide more insight into the strengths and weaknesses of the respective prudence methods and advance knowledge and application relevance of the methods. The original RDM method uses a Single Classification RDR engine. An intellectual contribution of this project is to develop a Multiple Classifications based RDM (MC-RDM) which is the version actually compared to RM.

A further contribution proposed by this project is the development of a novel prudent method from a merger of RM and MC-RDM. The new method; Integrated Prudent Analysis (IPA) is anticipated to use the collective strength of the two methods to eliminate or reduce the limitations of using each individual method. Most research in fraud detection is limited by a use of synthetic data or lack of real records from the appropriate domain(s). In contrast, the commercial contribution of this project is based on the application of IPA in Internet banking fraud detection using real online banking transactions. This is where this project's commercial relevance and industrial application sets it apart from other projects with a similar objective.

8.4 The Results

A host of tests and comparisons were conducted including comparing SC-RDM and MC-RDM; RM and RDM over eight datasets; RM, RDM and IPA; and finally, running IPA on Internet banking transactions. Comparisons were based on two main metrics, classifier accuracy (or simple accuracy) and prudence accuracy (or balanced accuracy).

Generally, RDM and RM showed good results, validating what had been reported about their performance previously (Prayote, 2007; Dazeley, 2007). According to the metrics used in this project, RDM was slightly better than RM in prudence accuracy in categorical datasets and RM had a slight advantage over RDM in numerical datasets. Over all the seven public datasets, the two systems performances are generally close and comparable to one another. In the end, this research concluded that given the two systems performance proximity to one

another across all the datasets, there could not be an outright best method as far as the prudence accuracy metric was concerned. It was further concluded that choosing whatever method was best for a given domain would be determined by each system's results in that domain. So far, RDM seems suited for categorical data and RM for numerical datasets.

A comparison of the two methods (RM and RDM) with IPA favoured the combined method in terms of prudence accuracy. This confirmed the project's hypothesis that a particular combination of the two methods could improve each method's individual prudence accuracy. IPA was further tested with Internet banking transactions showing good results. The application of IPA in Internet banking fraud detection was not necessarily meant to show-off the system's good prudence accuracy in this domain but rather to demonstrate the viability of prudence methods in fraud detection applications.

8.5 Contributions

In accordance with the projected goals, most of the project's objectives were met. Specifically, the following can be explicitly mentioned as contributions of the project.

- Redevelopment of Multiple Classifications RDM from Single Classification RDM. Using an MCRDR engine, a Multiple Classifications-RDM system was developed.
- Hitherto unpublished, focused comparisons of RM and RDM were conducted. A conference paper was published detailing results of preliminary tests of the two systems.
- Introduction of RDR prudence to Internet banking fraud detection. RDR had not been applied in the Internet banking fraud detection domain. A conference paper reporting early test results and explaining the viability of prudent fraud detection in online banking was published.
- Development of IPA by combining RM and RDM. Combining two known prudence methods RM and RDM, a more accurate prudent system IPA was developed.
- Application of IPA to Internet banking fraud detection. A commercial contribution and practical milestone of this project was when the newly developed IPA system was tested on Internet banking transactions.

8.6 Conclusion

In closing, it is the position of this project that the set objectives were met. A Multiple Classifications version of RDM was developed, a series of tests were run to compare RM and MC-RDM and IPA; a combination of RM and RDM was developed and shown to have better performance than any of the two systems. IPA was applied to online banking fraud detection, showing a good potential for an RDR based, prudent fraud detections system in this domain. A number of minor distractions worth flagging include the need for a unified, cross-domain and standardised KBS evaluation framework/standard. A number of disparate methods are being used in different domains to test and evaluate KBS. The existing methods evaluate different aspects of a KBS and usually vary from one domain to another. RM and RDM have a number of parameters, most of which directly affect the systems' performances. Having developed RM and MC-RDM from scratch, it became impossible for this project to test all possible RM, RDM and IPA thresholds and parameters. This issue is also highlighted by Dazeley (2007).

Future work in this field could organise a systematic way of organising systems' parameters and conduct further tests with unreported parameters. For some datasets, it was impossible to get perfect simulated experts from See5. Consequently, some tests were done with average SEs and others with fairly accurate SEs. Investigating the effect of different levels of SE accuracy on prudence would help explain some of the results reported in this project especially on the online banking data. There is still plenty of room to experiment with other configurations of IPA by trying different combinations of RM and RDM. The IPA configuration reported as the best of all the developed versions may not be so given the availability of other thresholds and parameters. Given the introduction and advances in new KBS evaluation methodologies, it may be worth evaluating IPA with the dynamic evaluation proposed by Beydoun and Hoffman (2013) and the run time validation approach of Finlayson and Compton (2013).

9.Bibliography

Australian Federal Police. (2012, May). Retrieved from Australian Federal Police: http://www.afp.gov.au/policing/cybercrime/internet-fraud-and-scams.aspx

FlowMatrix A. l. (2011, January 20). *FlowMatrix-Network Behavior Analysis System*. Retrieved January 21, 2011, from Xharru ltd- Home of FlowMatrix and Network Simulator: http://www.akmalabs.com/flowmatrix.php

Abraham, A. (2005). *Rule-Based Expert Systems.* John Wiley & Sons.

Abrazhevich, D. (2001). Classification and Characteristics of Electronic Payment Systems. *International Conference on Electronic Commerce and Web Technologies* , (pp. 81-90).

Androulidakis, I., & Papapetros, D. (2008). Survey Findings towards Awareness of Mobile Phones' Security Issues. *International Conference on Data Networks, Communications and Computers (DNCOCO)*, (pp. 130-135).

Australian Bureau of Statistics. (2010, June). Retrieved from Australian Bureau of Statistics: http://www.abs.gov.au/AUSSTATS/abs@.nsf/Lookup/1301.0Feature+Article13012009%E2%80%9310

ACI Worldwide. (2011, January 18). *ACI payment systems*. Retrieved January 18, 2011, from ACI Worldwide: http://www.aciworldwide.com/igsbase/igstemplate.cfm/SRC=DB/SRCN=/GnavID=15

Aggarwal, C., & Yu, P. (2001). Outlier Detection for High Dimensional Data. *2001 ACM SIGMOD International Conference on Management of Data* , (pp. 37-46). New York.

Aha, D. (1991). *Tic-Tac-Toe Endgame Data Set .* Retrieved 2012, from http://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame

Aladwani, A. (2001). Online banking: a field study of drivers, development challenges and expectations. *International Journal of Information Management*(21), 213-225.

Aleskerov, E., Freisleben, B., & Rao, B. (1997). Cardwatch: a neural network based database mining system for credit card fraud detection . *Computational Intelligence for Financial Engineering*, (pp. 173-200).

Anti-Phishing Working Group. (2010). *Phishing Activity Trends Report.* APWG.

Anti-Phishing Working Group. (2011). *Phishing Activity Trends Report.* APWG.

Asokan, N., Janson, P., Steiner, M., & Waidner, M. (1996). *Electronic Payment Systems.* IBM Research Division, Zurich.

Barnett, V., & Lewis, T. (1995). *Outliers in Statistical Data* (3 ed.). Wiley.

Barwise, P. (1997). Editorial. *The Journal of Brand Management*, 220-223.

Beale, R., & Jackson, T. (1991). *Neural Computing: An introduction.* Bristol, Great Britain: IOP Publishing.

Ben-Gal, I. (2005). Outlier Detection. In O. R. Maimon, *Data Mining and Discovery Handbook: A Complete Guide for Practitioners and Researchers* (pp. 131-146). Kluwer Academic Publishers.

Beydoun, G., & Hoffman, A. (2013). Dynamic evaluation of the development process of knowledge-based information systems. *Knowledge Information Systems, 35*, 233-247.

Bolton, R., & Hand, D. (2002). Statistical fraud detection: A Review . *Statistical Science, 17*(3), 235-249.

Booz, Allen, & Hamilton. (1997). *Internet Banking; a global study of potential.* New York: Booz, Allen & Hamilton Inc.

Breunig, M., Kriegel, H., Ng, R., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. *ACM SIGMOD*, (pp. 93-104).

Buchanan, B., & Shortliffe, E. (1984). *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project .* Reading: Addison-Wesley.

Cao, T., & Compton, P. (2005). A simulation framework for knowledge acquisition evaluation. *28th Australasian Computer Science Conference*, (pp. 353-360). Newcastle.

Car Advice. (2012, July). Retrieved July 26, 2012, from http://www.caradvice.com.au/140921/bmw-1-series-review/

Centor, R. (1991). Signal Detectablity: The Use of ROC Curves and Their Analyses . *Medical Decision Making*, 102-106.

Charnes, A., Cooper, W., & Rhodes, E. (1978). Measuring the Efficiency of Decision Making Units. *European Journal of Operational Research, II*(6), 429-444.

Chen, W.-H., Hsu, S.-H., & Shen, H.-P. (2005). Application of SVM and ANN for intrusion detection. *Elsevier: Computers and OPerations Research*, 2717-2634.

Compton, P., & Cao, T. (2006). Evaluation of Incremental Knowledge Acquisition with Simulated Experts. *Australian Conference on Artificial Intelligence*, (pp. 39-48).

Compton, P., & Horn, K. (1989). Maintaining an Expert System. *Applications of Expert Systems*, 366-385.

Compton, P., & Jansen, R. (1988). Knowledge in context: a strategy for expert system maintenance. *Australian Joint Conference on Artificial intelligence* (pp. 292-306). Springer-Verlag New York.

Compton, P., Kang, B., Preston, P., & Mulholland, M. (1993). Knowledge Acquisition Without Analysis. In *Knowledge Acquisition for Knowledge Based Systems* (pp. 278-299). Berlin: Springer Verlag.

Compton, P., Peters, L., Edwards, G., & Lavers, T. (2005). Experience with Ripple-Down Rules. *Artificial Intelligence-2005.* Cambridge.

Compton, P., Preston, P., & Kang, B. (1995). The Use of Simulated Experts in Evaluating Knowledge Acquisition . *The 9th Knowledge Acquisition for Knowledge Based Systems Workshop*, (pp. 12- 30). Calgary.

Compton, P., Preston, P., Edwards, G., & Kang, B. (1996). Knowledge Based Systems That Have Some Idea of Their Limits. *CIO, 15*, 57-63.

Datamonitor. (2009). *Security in Online Banking (Strategic Focus).* DataMonitor.

Dayhoff, J. E., & DeLeo, J. M. (2001). Artificial Neural Networks- Opening the Black Box. *CANCER Supplement*, 1615-1635.

Dazeley, R. (2007). *To the Knowledge Frontier and Beyond: A Hybrid System for Incremental Contextual-Learning and Prudence Analysis.* PhD Thesis, University of Tasmania.

Dazeley, R., & Kang, B. (2008). Detecting the Knowledge Boundary with Prudence Analysis. *AI 2008*, (pp. 482-488). Auckland.

Dazeley, R., & Kang, B. (2008). The Viability of Prudence Analysis. *The Pacific Rim Knowledge Acquisition Workshop*, (pp. 107-121). Hanoi.

Dazeley, R., Warner, P., Johnson, S., & Vamplew, P. (2010). The Ballarat Incremental Knowledge Engine. *Pacific Knowledge Acquisition Workshop (PKAW)* (pp. 195-207). Springer Link.

Dazeley, R., Park, S., & Kang, B. (2011). Online knowledge validation with prudence analysis in a document management application. *Expert Systems with Applications, 38*, 10959-10965.

Detica. (2011). *The Cost Of Cybercrime.* Detica Ltd, Surrey.

Ditcheva, B., & Fowler, L. (2005). *Signature-based Intrusion Detection.* Chapel Hill: University of North Carolina.

Durkin, J. (1994). *Expert systems: design and development.* Macmillan.

Edwards, G., Compton, P., Malor, R., Srinivasan, A., & Lazarus, L. (1993). PEIRS: a pathologist maintained expert system for the interpretation of chemical pathology reports. *Pathology, 25*(1), 27-34.

Edwards, G., Kang, B., Preston, P., & Compton, P. (1995). Prudent Expert Systems with Credentials: Managing the expertise of decision support systems. *International Journal of Bio-Medical Computing, 40*, 125-132.

Elkan, C., & Greiner, R. (1993). Book Review: Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project . *Artificial Intelligence*, 41-52.

Enterprise Management Associates. (2012). *The Industrialization of Fraud Demands a Dynamic Intelligence-Driven Response.* Boulder: Enterprise Management Associates.

Fawcett, T. (2003). *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers.* Palo Alto: HP Laboratories.

Fawcett, T., & Provost, F. (1997). Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*(1), 291-316.

Feigenbaum, E., & Buchanan, B. (1978). Dendral and Meta-Dendral: Their Applications Dimension. *Artificial Intelligence*, 5-24.

Feigenbaum, E., & Buchanan, B. (1993). DENDRAL and Meta-DENDRAL: roots of knowledge systems and expert system applications. *Artificial Intelligence*, 233-240.

Federal Financial Institutions Examination Council. (2012). *Authentication in an Internet Banking Environment.* Arlington: FFIEC.

FICO. (2011, January 18). *Falcon Fraud Manager.* Retrieved January 18, 2011, from FICO: http://www.fico.com/en/products/dmapps/pages/fico-falcon-fraud-manager.aspx

Finlayson, A., & Compton, P. (2013). Run-time validation of knowledge-based systems. *International Conference on Knowledge Capture .* Banff.

Gaines, B. (2000). Knowledge Science and Technology: Operationalizing the Enlightenment. *6th Pacific Knowledge Acquisition Workshop.* Sydney.

García, V., Mollineda, R. A., & Sánchez, J. S. (2009). Index of Balanced Accuracy: A Performance Measure for Skewed Class Distributions. *Lecture Notes in Computer Science* , pp. 441-448.

Gholamreza, N., & Schnabl, A. (1997). Development of Multi-Criteria Metrics for Evaluation of Data Mining Algorithms. *Knowledge Discovery and Data Mining*, (pp. 37-42). Newport Beach.

Ghosh, S., & Reilly, D. (1994). Credit Card Fraud Detection with a Neural-Network. *27th International Conference on System Sciences* , (pp. 621-630). Hawaii.

Giarratano, J., & Riley, G. (2005). *Expert Systems: Principles and Programming.* Thomson Course Technology.

Gonzalez, F. A., & Dasgupta, D. (2003). Anomaly Detection Using Real-Valued Negative Selection. *Genetic Programming and Evolvable Machines*, pp. 383-403.

Grogono, P., Preece, A., Shingal, R., & Suen, C. (1993). *A Review of Expert Systems Evaluation Techniques.* Technical Report, Concordia University, Montreal.

Guida, G., & Mauri, G. (1993). Evaluating Performance and Quality of Knowledge-Based Systems: Foundation and Methodology. *IEEE Transactions in Knowledge and Data Engineering*, 204-224.

Guo, T., & Li, G.-Y. (2008). Neural Data Mining for Credit Card Fraud Detection . *7th International Conference on Machine Learning and Cybernetics*, (pp. 3630-3634). Kunming.

Guo, Y., Heflin, J., & Pan, Z. (2003). Benchmarking DAML+OIL Repositories. *International Semantic Web Conference*, (pp. 613-627).

Guo, Y., Pan, Z., & Heflin, J. (2004). An evaluation of knowledge base systems for large OWL datasets . *International Semantic Web Conference*, (pp. 278-288).

Gupta, U. (1991). *Validating and verifying knowledge-based systems.* Los Alamitos: IEEE Computer Society Press.

Han, S., Mirowski, L., Jeon, S. H., Lee, G. S., Kang, B., & Turner, P. (2013). Expert Systems and Home-based Telehealth: Exploring a role for MCRDR in enhancing diagnostics. *Advanced Science and Technology Letters, 22*, 121-127.

Hardison, N., Reif, D., Fanelli, T., Ritchie, M., Dudek, S. M., & Motsinger-Reif, A. (2008). A Balanced Accuracy Fitness Function Leads to Robust Analysis using Grammatical Evolution Neural Networks in the Case of Class Imbalance . *GECCO'08*, (pp. 353-354).

Havinga, P., Gerard, J., & Smit, A. (1996). Survey of Electronic Payment Methods and Systems. *Memoranda Informatica*, 7-28.

Hawkins, D. (1980). Identification of Outliers. *Biometrical Journal, 29*(2), 188-198.

Hayes-Roth, F. (1985). Rule-Based Systems. *Communications of the ACM*, 921-932.

Hayes-Roth, F., & Jacobstein, N. (1994). The State of Knowledge-Based Systems. *Communications of the ACM*, 27-39.

Hodge, J., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review, 22*, 85-126.

International Business Machines. (2008). *Improving Payments Fraud Detection and Prevention: ACI Proactive Risk Manager with IBM System z10.* IBM Corporation. Somers: IBM Corporation.

Joffe, E., Havakuk, O., Herskovic, J., Patel, V., & Bernstam, E. V. (2012). Collaborative knowledge acquisition for the design of context-aware alert system. *Journal of American Medical Informatics Association, 19*, 988-994.

Jones, A., & Sielken, R. (2000). *Computer System Intrusion Detection: A Survey.* Technical Report, University of virginia.

Kabiri, P., & Ghorbani, A. (2005, September). Research on Intrusion Detection and Response: A Survey. *International Journal of Network Security, 1*(2), 84-102.

Kang, B., Compton, P., & Preston, P. (1995). Multiple Classification Ripple Down Rules: Evaluation and Possibilities. *9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop*, (pp. 17-26). Banff.

Kazienko, P., & Dorosz, P. (2004). *Intrusion Detection Systems (IDS) Part 2 - Classification; methods; techniques.* Retrieved March 26, 2010, from WindowSecurity.com: http://www.windowsecurity.com/articles/IDS-Part2-Classification-methods-techniques.html?

Kou, Y., Lu, C., Sirwongwattana, S., & Huang, Y. (2004). Survey of Fraud Detection Techniques. *International Coneference on Networking, Sensing and Control*, (pp. 749-754). Taipei.

Kriegel, H. P. (2012). *Density-Based Cluster- and Outlier Analysis.* Retrieved from University of Munich Insitute for Computer Science: http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/index.html

Krivko, M. (2010). A Hybrid Model for Plastic Card Fraud Detection Systems. *Expert Systems with Applications, 37*, 6070-6076.

Kvarnstrom, H., Lundin, E., & Jonsson, E. (2000). Combining fraud and intrusion detection-meeting new requirements. *Nordic Workshop on Secure IT Systems*, (pp. 11-19).

Last, M., & Kandel, A. (2001). Automated Detection of Outliers in Real-World Data. *Second International Conference on Intelligent Technologies*, (pp. 292-301).

Li, B., Xie, S., & Xu, X. (2011). Recent development of knowledge-based systems, methods and tools for One-of-a-kind Production. *Knowledge Based Systems*, 1108-1119.

Liao, S. (2003). Knowledge management technologies and applications- literature review from 1995 to 2002. *Expert Systems with Applications*, 155-164.

Lindsay, R., Buchanan, B. G., Feigenbaum, E., & Lederberg, J. (1993). A Case Study of the First Expert System for Scientific Hypothesis Formation. *Artificial Intelligence - AI*, 209-261.

Lippmann, R., Fried, D., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Webster, S., Wyschogrod, D., Cunninghan, R., Zissman, M. (2000). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. *DARPA Information Survivability Conference*, (pp. 12-26).

Liu, L., & Liang, Q. (2011). A high-performing comprehensive learning algorithm for text classification without pre labeled training set. *Knowledge Information Systems, 29*, 727-738.

Macquarie Australia's National Dictionary (3rd ed.). (2001). Sydney, Australia: Macquarie Library.

Mannan, M., & Oorschot, P. (2007). Security and usability: the gap in real-world online banking. *Workshop on New Security Paradigms* , (pp. 1-14). New Hampshire.

Marsland, S. (2003). Novelty detection in learning systems. *Neural Computing Surveys*, 157-195.

Maruatona, O., Vamplew, P., & Dazeley, R. (2012). Prudent Fraud Detection in Internet Banking. *Cybercrime and Trustworthy Computing Workshop 2012.* Ballarat.

Maruatona, O., Vamplew, P., & Dazeley, R. (2012). RM and RDM, a Preliminary Evaluation of two Prudent RDR Techniques. *The Pacific Rim Knowledge Acquisition Workshop*, (pp. 188-194). Kuching.

McCall, T. (2012, May). Retrieved from Gartner Inc.: http://www.gartner.com/it/page.jsp?id=565125

McCombie, S. (2008). Trouble in Florida, The Genesis of Phishing attacks on Australian Banks. *6th Australian Digital Forensics Conference.* Perth.

Mendenhall, W., Reinmuth, J., & Beaver, R. (1993). *Statistics for Management and Economics.* Belmont: Duxbury Press.

Metz, C. (1978). Basic Principles of ROC Analysis. *Seminars in Nuclear Medicine*, 283-298.

Mhamane, S., & Lobo, L. (2012). Internet banking fraud detection using HMM. *Computing Communication & Networking Technologies (ICCCNT)*, (pp. 1-4). Coimbatore.

Mihaela, O. (2006). On the Use of Data-Mining Techniques in Knowledge-Based Systems. *Economy Informatics, VI*(1-4), 21-24.

Ng, S. K., McLachlan, G. J., & Lee, A. H. (2006). An incremental EM-based learning approach for on-line prediction of hospital resource utilization. *Artificial Intelligence in Medicine, 36*, 257-267.

Nilsson, M., Adams, A., & Herd, S. (2005). Building security and trust in online banking. *Conference on Human Factors in Computing Systems*, (pp. 1701-1704). New York.

Patel, A., Qassim, Q., & Wills, C. (2010). A survey of intrusion detection and prevention systems. *Information Management and Computer Security, 18*(4), 277-290.

Phua, C., Alahakoon, D., & Lee, V. (2004). Minority Report in Fraud Detection: Classification of Skewed Data. *SIGKDD Explorations*, pp. 50-59.

Phua, C., Lee, V., Smith, K., & Gayler, R. (2005). A Comprehensive Survey of Data Mining-based Fraud Detection Research. *Artificial Intelligence Review*.

Polika, R., Udpa, L., Udpa, S., & Honavar, V. (2004). An incremental learning algorithm with confidence estimation for automated identification of NDE signals. *Transactions on Ultrasonics, Ferroelectrics and Frequency Control, 51*, 990-1001.

Powers, R., Goldszmidt, M., & Cohen, I. (2005). *Short Term Performance Forecasting in Enterprise Systems.* Stanford: Stanford University.

Prayote, A. (2007). *Knowledge Based Anomaly Detection.* PhD Thesis, University of New South Wales, Sydney.

Prayote, A., & Compton, P. (2006). Detecting Anomalies and Intruders. *International Conference on Artificial Intelligence 2006*, (pp. 1084-1088). Hobart.

Preece, A. (2001). Evaluating Verification and Validation Methods in Knowledge Engineering. *Micro-Level Knowledge Management* , 123-145.

Putland, P., & Hill, J. (1997). Electronic payment systems. *BT Technology Journal, 15*(2), 32-38.

Pyle, D. (1999). *Data preparation for Data Mining.* San Francisco: Morgan Kaufmann.

Reserve Bank of India. (2001, June). *Report on Internet Banking.* Retrieved from Reserve Bank of India: http://www.rbi.org.in/scripts/BS_ViewPublicationReport.aspx

Richards, D. (2003). Knowledge-Based System Explanation: The Ripple-Down Rules Alternative. *Knowledge and Information Systems*, 2-25.

Richards, D. (2009). Two decades of Ripple Down Rules research. *The Knowledge Engineering Review, 24*(2), 159-184.

RSA Security. (2010). *2010 Special Online Fraud Report.* White Paper, RSA Security Inc.

RSA Security. (2011). *Sophisticated Local Pharming Trojan Targets Brazilian Banks.* RSA.

RSA Security. (2011). *The Psychology of Social Engineering.* RSA.

RSA Security. (2012). *Online Fraud Report.* RSA.

Rulequest. (2012, August). Retrieved from RuleQuest data mining tools: http://www.rulequest.com/see5-info.html

Sadeghi, A., & Schneider, M. (2003). Electronic Payment Systems. In *Digital Rights Management* (pp. 113-137). Springer Berlin / Heidelberg.

Salim, M., Villavicencio, A., & Timmerman, M. (2003, January). A Method for Evaluating Expert System Shells for Classroom Instruction. *Journal of Industrial Technology, 19*(1), 2-11.

SAS. (2007). *SAS Fraud Management.* Technical Report, SAS Institute Inc.

Sathye, M. (1999). Adoption of Internet banking by Australian consumers: an empirical investigation. *International Journal of Bank Marketing*, 324-334.

SECTOOLS. (2011, January 21). *Top 5 Intrusion Detection Systems*. Retrieved January 21, 2011, from SECTOOLS.ORG: http://sectools.org/ids.html

Senator, T. (2009). On The Efficacy of Data Mining for Security Applications. *Cyber Security and Intelligence-Knowledge Discovery and Data mining* , (pp. 75-83). Paris.

Smaha, S. (1988). Haystack: an intrusion detection system. *Aerospace Computer Security Applications Conference*, (pp. 37-44).

Smith, R. (2001). *Internet Related Fraud: Crisis or Beat-up?* Australian Institute of Criminology, Canberra.

SNORT. (2011, January 21). *SNORT:: Home Page*. Retrieved January 21, 2011, from SNORT: http://www.snort.org/

Sood, A., & Enbody, R. (2011). The state of HTTP declarative security in online banking websites. *Computer Fraud and Security*, 11-16.

Spackman, K. (1989). Signal Detection Theory: Valuable Tools for evaluating inductive learning. *International Workshop on Machine Learning*, (pp. 160-163). San Mateo.

Stolfo, S., Fan, D., Lee, W., & Prodromidis, A. (1997). *Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results.* AAAI.

Stolfo, S., Fan, W., Lee, W., Prodromidis, A., & Chan, P. (2000). Cost-based modeling for fraud and intrusion detection: results from the JAM project. *DARPA Information Survivability Conference and Exposition*, (pp. 130-144).

Sweets, J., Dawes, R., & Monahan, J. (2000). Better decisions through science. *Scientific American*, 82-87.

University of California Irvine. (2012, June). Retrieved from UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/index.html

Weatherford, M. (2002). Mining for Fraud. *IEEE Intelligent Systems*, pp. 4-5.

Weber, R., & Darbellay, A. (2010). Legal issues in Mobile Banking. *Journal of Banking Regulation*, 129-145.

Wei, W., Li, J., & Cao, L. (2012). Effective Detection of Sophisticated Online Banking Fraud on Extremely Imbalanced Data. *World Wide Web, 16*, 449-475.

Wiig, K. (1994). *Knowledge management: the central management focus for intelligent-acting organizations.* Arlington: Schema Press.

Wolpert, D., & Macready, W. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation* , 67-82.

Yoon, H., Han, S., Kang, B., & Park, S. (2012). V & V to use agile approach in ES development: Why RDR works for expert system developments! *Communications in Computer and Information Science, 340*, 113-120.

Appendix A: Acronyms

AD: Anomaly detection

AIC: Australian Institute of Criminology

ANN: Artificial Neural Network

BA: Balanced Accuracy

CDNLF: Continuously Differentiable non-liner Function

DEA: Data Envelopment Analysis

DMU: Decision Making Units

EPS: Electronic Payment System

ES: Expert Systems

FEP: Feature Exception Prudence

FFIEC: Federal Financial Institutions Examination Council

FPA: Function Point Analysis

FRP: Feature Recognition Prudence

HMM: Hidden Markov Model

ID: Intrusion Detection

IPA: Integrated Prudent Analysis

KB: Knowledge-Base

KBS: Knowledge-Based Systems

MCRDR: Multiple Classifications Ripple Down Rules

MD: Misuse Detection

NNIDS: Network Node Intrusion Detection Systems

OBS: Online Banking System

OCC: One Class Classification

OD: Outlier Detection

OEBA: Outlier Estimation with Backward Adaptability

OECA: Outlier Estimation for Categorical Attributes

OWL: Web Ontology Language

PIERS: Pathology Interpretative Expert Reporting System

RBF: Radial Basis Function

RBS: Rule Based Systems

RDM: Ripple Down Models

RDR: Ripple Down Rules

RM: Rated Multiple Classifications Ripple Down Rules

ROC: Receiver Operating Characteristics

SP: Situated Profile

V&V: Verification and Validation