Fraud Detection for Online Banking for Scalable and Distributed Data

IKRAM UL HAQ

Thesis

Submitted in total fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Science, Engineering and Information Technology
Federation University Australia

PO Box 663
University Drive, Mount Helen
Ballarat Victoria 3353
Australia

November 2019

To my parents & family
and
My supervisors Prof Iqbal Gondal and A/Prof Peter Vamplew

Abstract

Online fraud causes billions of dollars in losses for banks. Therefore, online banking fraud detection is an important field of study. However, there are many challenges in conducting research in fraud detection. One of the constraints is due to unavailability of bank datasets for research or the required characteristics of the attributes of the data are not available. Numeric data usually provides better performance for machine learning algorithms. Most transaction data however have categorical, or nominal features as well. Moreover, some platforms such as Apache Spark only recognizes numeric data. So, there is a need to use techniques e.g. One-hot encoding (OHE) to transform categorical features to numerical features, however OHE has challenges including the sparseness of transformed data and that the distinct values of an attribute are not always known in advance. Efficient feature engineering can improve the algorithm's performance but usually requires detailed domain knowledge to identify correct features.

Techniques like Ripple Down Rules (RDR) are suitable for fraud detection because of their low maintenance and incremental learning features. However, high classification accuracy on mixed datasets, especially for scalable data is challenging. Evaluation of RDR on distributed platforms is also challenging as it is not available on these platforms.

The thesis proposes the following solutions to these challenges:

- We developed a technique Highly Correlated Rule Based Uniformly Distribution (HCRUD) to generate highly correlated rule-based uniformly-distributed synthetic data.
- We developed a technique One-hot Encoded Extended Compact (OHE-EC) to transform categorical features to numeric features by compacting sparse-data even if all distinct values are unknown.

- We developed a technique Feature Engineering and Compact Unified Expressions (FECUE) to improve model efficiency through feature engineering where the domain of the data is not known in advance.
- A Unified Expression RDR fraud deduction technique (UE-RDR) for Big data has been proposed and evaluated on the Spark platform.

Empirical tests were executed on multi-node Hadoop cluster using well-known classifiers on bank data, synthetic bank datasets and publicly available datasets from UCI repository. These evaluations demonstrated substantial improvements in terms of classification accuracy, ruleset compactness and execution speed.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due references are made in the text of the thesis.

IKRAM UL HAQ
Nov, 2019

Publications

- (Ul Haq, Gondal, Vamplew, & Layton, 2016). Generating Synthetic Datasets for Experimental Validation of Fraud Detection, The 14th Australasian Data Mining Conference, Canberra, Australia, 2016

- (Ul Haq, Gondal, Vamplew, & Brown, 2018). Categorical Features Transformation with Compact One-hot Encoder for Fraud Detection in Distributed Environment, The 16th Australasian Data Mining Conference, Bathurst, NSW, Australia, 2019

- (Ul Haq, Gondal, & Vamplew, 2019). Enhancing Model Performance for Fraud Detection by Feature Engineering and Compact Unified Expressions, ICA3PP 2019 - 19th International Conference on Algorithms and Architectures for Parallel Processing, Melbourne, Australia, 2019

- (Ul Haq, Gondal, & Vamplew, 2020). Unified Expression Ripple Down Rules based Fraud Detection Technique for Scalable Data, AISC 2020 - Australasian Information Security Conference, Melbourne, Australia 2020 (Submitted)

- (Ul Haq et al., 2020). Fraud Detection Techniques for Scalable and Distributed Data – Journal of Computers & Security (Ready for Submission)

General Declaration – Published works

I hereby declare that this thesis does not contain any material that has been accepted for the award of any other degree or diploma at a university institution. I also state that, to the best of my knowledge and belief, this thesis does not contain any material previously published or written by another person, except where due references are made in the text of the thesis.

Four original papers (published) were part of this research in peer-reviewed ranked conferences. I was fully responsible for developing and writing all the papers in the thesis under the supervision of Prof Iqbal Gondal and A/Prof Peter Vamplew.

Following table gives the level of my contribution for chapters 3, 4, 5 and 6:

Thesis Chapter	Publication Title	Publication Status	Contribution	Contribution %
3	Generating Synthetic Datasets for Experimental Validation of Fraud Detection	Published	Developed analytical model and experimental setup. I wrote the initial draft and incorporated the suggestions & recommendations from the supervisors to prepare the final draft	80%
4	Categorical Features Transformation with Compact One-hot Encoder for Fraud Detection in Distributed Environment	Published	Developed analytical model and experimental setup. I wrote the initial draft and incorporated the suggestions & recommendations from the supervisors to prepare the final draft	80%
5	Enhancing Model Performance for Fraud Detection by Feature Engineering and Compact Unified Expressions	Published	Developed analytical model and experimental setup. I wrote the initial draft and incorporated the suggestions & recommendations from the supervisors to prepare the final draft	80%
6	Unified Expression Ripple Down Rules Based Fraud Detection Technique for Scalable Data	Submitted	Developed analytical model and experimental setup. I wrote the initial draft and incorporated the suggestions & recommendations from the supervisors to prepare the final draft	80%

Student Signature: Date:

The undersigned hereby certify that the above declaration correctly reflects the nature and context of the student and co-authors contribution to this work.

Principal Supervisor Signature: Date:

Acknowledgments

At this very special moment, first of all, I would like to express my heartfelt gratitude to the Almighty Allah for allowing me to successfully complete this PhD study. I am grateful for the tremendous blessings that the Almighty has showered me not just during my research, but throughout my life. To achieve this big objective, I have been able to consult and get help from others, so I would like to extend my deepest appreciation to colleagues who have assisted in completing this thesis. First of all, I would like to express my sincere appreciation for the patience and guidance of my research supervisors Prof Iqbal Gondal and A/Prof Peter Vamplew throughout the study. Special thanks to Dr R. Layton, Dr O. Maruatona and Dr B. Venki for guiding me in the initial days of my research. I gained significant insights through my meetings with my supervisors to conduct research in a strategic manner. They provided me with timely advice and I benefited from their experience significantly. I am also grateful to Prof. J. Abawajy and Dr. S. Huda from Deakin University for their support at the start of my PhD. I want to express my sincere thanks to Credit Objects Pty Ltd, Australia for the moral support for my research. I would like to thank the Internet Commerce Security Lab (ICSL) – Federation University team for their help and encouragement, especially Helen Wade. Thanks to the senior management at ANZ Bank, Australia, who gave me a chance to apply my research to the financial sector datasets containing billions of the records. I would also like to thank the Federation University for offering the opportunity to conduct research in ICSL, providing excellent digital Library resources and providing excellent IT and research support. Special thanks to NECTAR Research Cloud for providing computing infrastructure, software and services for setting up a multi-node Hadoop cluster.

I am extremely grateful to my parents and my whole family for their sacrifice and care throughout my life. Their unconditional love and never-ending affection gave me the confidence to use my skills and knowledge. Special thanks to my wife for taking care of the kids while I was busy in my research. My parents always encouraged me throughout my life, and every success in my life is attributed to them. I want to thank my children for their beautiful smiles and family members for their endless support and inspiration. I would

like to acknowledge that my study was supported by an Australian Government Research Training Program (RTP) Fee-Offset Scholarship through Federation University Australia.

Finally, I want to thank all of you sincerely for being with me to make this work a success.

IKRAM UL HAQ

November 2019

Table of Contents

Table of Contents

List of Figures

List of Tables

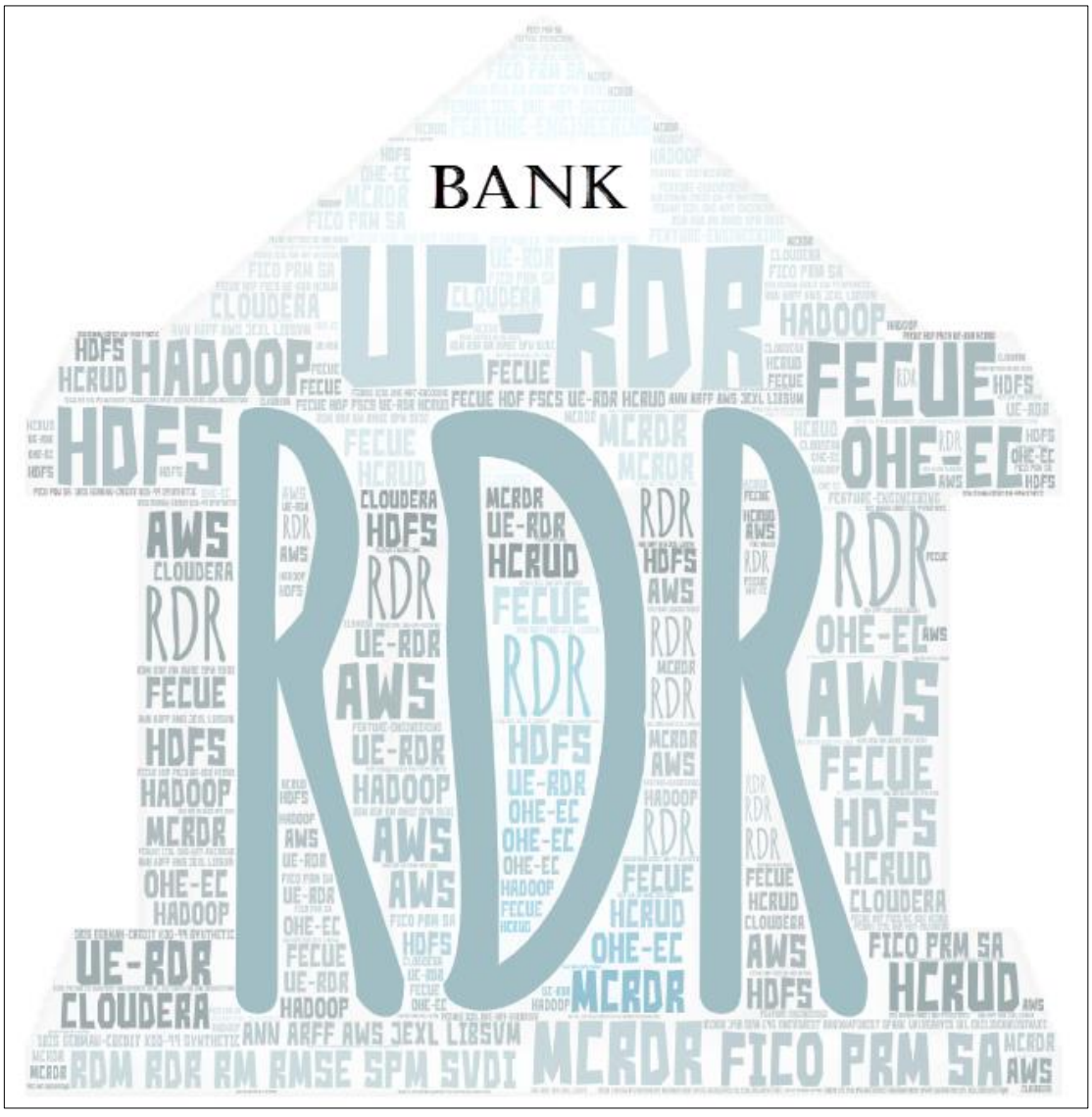
Acronyms

AD	Anomaly Detection
ANN	Artificial Neural Network
API	Application Program Interface
ARFF	Attribute-Relation File Format
AIS	Artificial Immune Systems
AWS	Amazon Web Services
CDE	Coupled Data Embedding
CSV	Comma Separated Values
DAG	Directed Acyclic Graph
DL	Deep Learning
ES	Expert Systems
FD	Fraud Detection
FE	Feature Engineering
FECUE	Feature Engineering and Compact Unified Expressions
HCRUD	Highly Correlated Rule Based Uniformly Distribution
HDFS	Hadoop Distributed File System
IBL	Instance-Based Learning
IC3	Internet Crime Complaint Centre
ICSL	Internet Commerce Security Laboratory
ID	Intrusion Detection
IDS	Intrusion Detection Systems
IPA	Integrated Prudence Analysis
JEXL	Java Expression Language
KA	Knowledge Acquisition
KB	Knowledge Base
KBS	Knowledge Based Systems

ML	Machine Learning
MCRDR	Multiple Classifications Ripple Down Rules
MCSI	Microsoft Computing Safety Index
MD	Misuse Detection
MVI	Missing Value Imputation
NECTAR	Network for Effective Collaboration Technologies Through Advanced Research
OBS	Online Banking System
OD	Outlier Detection
OHE	One-hot Encoding
OHE-EC	One-hot Encoded Extended Compact
PA	Prudence Analysis
PRM	Proactive Risk Manager
RDD	Resilient Distributed Dataset
RDM	Ripple Down Model
RDR	Ripple Down Rules
RM	Rated-MCRDR
RMSE	Root Mean Squared Error
RBS	Rule-based System
SD	Signature Detection
SP	Situated Profile
SPM	Situated Profile Models
SVDI	Singular Value Decomposition Interpolation
SVM	Support Vector Machine
UE	Unified Expressions
UEL	Unified Expression Language
UE-RDR	Unified Expression Ripple Down Rules Based Fraud Detection

Chapter 1

Introduction



Accurate fraud detection (FD) can enhance the confidence of the bank clients in doing their banking online. Fraud increases dramatically with the advancement of technology, which leads to significant losses for companies, so the identification of fraud has become a significant problem to investigate (Kou, Lu, Sirwongwattana, & Huang, 2004). It has been reported that fraudulent card transactions only in the United States cost $790 million in 2005 (Brabazon, Cahill, Keenan, & Walsh, 2010). The Microsoft Computer Safety Index survey revealed that the global annual impact of phishing and different forms of identity theft could amount to US$ 5 billion, whereas the cost of repairing damage to the reputation of online individuals could be as much as US$ 6 billion or on average an estimated US$ 632 for each loss (Marican & Lim, 2014). On the bases of crime complaints received by Internet Crime Complaint Centre (IC3)(FBI, 2018), IC3 has reported 161% increase in the loses in 2018.

The Oxford English Dictionary (Oxford, 2011) has defined fraud as “wrongful or criminal deception intended to result in financial or personal gain”. Australia Police Force (APF, 2018) defines Internet banking fraud as “fraud or theft committed using online technology to illegally remove money out of your account”. Commercial financial institutions have similar trends in the detection of fraud in online transactions as reported in the literature. By analysing various white papers and reports for commercial payment FD systems, (Hafiz, Aghili, & Zavarsky, 2016; Maruatona, 2013) concludes that the use of a rules-based scheme coupled with an Artificial Neural Network (ANN) strategy implemented for internet transaction FD is very suitable. Some of the examples of FD systems used in commercial OBS (ACI, 2011; FICO, 2010; Hafiz et al., 2016; Kount, 2006; SAS, 2007) are the FICO Application Fraud Manager, Proactive Risk Manager (PRM), the SAS Fraud Manager and Kount Fraud Prevention System. (Kou et al., 2004; Ngai, Hu, Wong, Chen, & Sun, 2011; Phua, Lee, Smith, & Gayler, 2010; Sharma & Panigrahi, 2013) have surveyed on FD approaches based on artificial immune systems (AIS), artificial intelligence, auditing, distributed and parallel computing, econometrics, expert systems (ES), fuzzy logic, genetic algorithms, machine learning (ML), neural networks, pattern recognition, statistics and visualisation. AIS is a recent artificial intelligence branch based on the human immune system's biological analogy (Brabazon et al., 2010).

Rather than just preventing the unauthorised transaction, internet banking FD systems also need to detect frauds immediately within a compromised account. (Richards, 2003) has indicated that conventional rule-based approach to knowledge acquisition (KA) are too slow, labour intensive and costly for a business. Brittleness was also one of the shortcomings in conventional rule-bases and these systems always attempt to give an answer even if it may be inaccurate. Brittleness refers to software that can be wrong when faced with an unpredictable situation. So, it is less accurate as it always trusts its current knowledge even for the cases where the knowledge is insufficient.

Deep Learning (DL) is new research field and is a machine learning branch where ANN learns from large of datasets (Goodfellow, Bengio, & Courville, 2016; Heaton, Polson, & Witte, 2016). DL needs no engineering and it extracts features automatically from raw data(Roy et al., 2018). DL succeeded tremendously in many areas of machine learning, however it has a number of limitations (Altexsoft, 2017; Chauhan & Singh, 2018; W. Chen, 2016), which include: High memory consuming, Need of very large data to train model, Need of very high computational power and lack of interpretability.

The Ripple Down Rules (RDR) strategy to KA has demonstrated significant benefits over standard rules. One such benefit is in terms of addition and removal of the rules due to the dynamicity of fraud environment. The RDR methods have shown that their rule addition and maintenance is better, faster and less expensive than conventional rule-bases. Prudence in RDR has been implemented to tackle the issue of brittleness. For Internet banking, prudent FD schemes mean precise and fast-tracking of new fraud trends, saving both financial institutions and clients' time, human resources and money. However, higher accuracy, compactness of the model, scalability and heterogeneous nature of data are some of the challenges to address in RDR. RDR implementation on distributed and scalable ML platform can be used to compare newly designed classifier with other scalable classifiers to study the performance in terms of FD.

This thesis presents studies done on FD for online banking and solves some challenges in this area. In the course of this research project, four conference papers were published in

different international conferences. The publications represent systematic research progression during the course of investigation in FD, and studies are presented in the form of chapters in this thesis. Overall, the thesis presents the background of the research problem, in-depth analysis of research challenges, details of the research methodology, followed by the findings and conclusions.

1.1 Research Problem

FD for online banking is a very significant field of studies, as cybercriminals devise sophisticated new fraud attacks on a daily basis, so this requires researchers to develop new FD techniques continually. According to (Maruatona, 2013) availability of detailed information regarding the FD system is very restricted as the banking industry rarely release FD statistics. In particular, the security providers of financial institutions are third party companies which also protect their intellectual property against their competitors. Therefore, both banks and IT agencies do not release their security systems information. (Bolton & Hand, 2002) also emphasize that developing new methods of identification of fraud is hard as the interchange of thoughts on the identification of fraud is very restricted, but the authors also promote the concept that techniques for detecting fraud should not be outlined publicly with details, otherwise criminals may benefit from the same data. (Phua et al., 2010) recognize that there are often two main criticisms of data mining for fraud identification: the scarcity of publicly available real experimental information, and the lack of well-publicized techniques and methods.

Bank datasets are needed to perform studies on fraud research. Banks sometimes provide information, but the information provided by the banks are either low in quantity or it may not have required features to validate new methods.

Heterogeneous nature of bank transactions data poses a challenge, i.e. a combination of numeric as well as mixed attributes in developing efficient FD techniques. ML algorithms work effectively on homogeneous data, either numeric or categorical data (Shih, Jheng, & Lai, 2010). The k-means based methods are efficient in processing large datasets but often are restricted to numerical data (Z. Huang, 1997). (K. Zhang & Jin, 2010) highlight that

existing outlier detection (OD) techniques are ineffective for mixed datasets. However, many ML problems have mixed features, rather than numeric features only. Moreover, some ML platforms, like Apache Spark, only accept numerical data. One-hot Encoding (OHE) (Harris & Harris, 2012; Wikipedia, 2017) technique is widely used to transform categorical features into numerical features in traditional data mining tasks. However, the sparseness of the transformed data with OHE and the fact that the distinct values of the attributes are not always known in advance; this presents a challenge to the OHE approach.

Classification accuracy and compactness of the model are very critical in FD. RDR is ideal among the existing rule-based techniques for fraud detecting due to its lower maintenance requirements and support for incremental learning (Compton & Jansen, 1988; Kelarev, Dazeley, Stranieri, Yearwood, & Jelinek, 2012; Richards, 2009). However, the performance of RDR on distributed and Big data platforms, in particular, Spark, has not been studied as RDR is not available on these platforms.

1.2 Research Objectives

Online banking FD ecosystem is very dynamic in nature and can be studied from various aspects. This thesis presents approaches that can detect fraudulent transactions very efficiently from large-scale and distributed datasets.

This research will address the following research objectives.

- **Objective 1 Synthetic Data Generation:**

  Fraud analysis research, especially for scalable data, needs a large size of bank transactions data. Either banks do not provide the data or the data provided by the bank is small, or they can not satisfy certain characteristics required to validate new methods and algorithms. The objective is to generate synthetic data from limited reference data from the bank. The developed technique is generic and can be applied to any datasets. The generated data must be labelled, have a high correlation to reference data, keep uniform distribution and maintain the same characteristics in generated data as in reference data. The uniform distribution must be true for individual attributes, the combination of attributes and the class labels as well.

- **Objective 2 Categorical Features Transformation for Fraud Detection in Distributed Environment:**

  ML algorithms provide improved efficiency on numeric data formats. But many datasets have categorical or nominal characteristics. For example, bank datasets are heterogeneous due to the nature of transactions data. The objective is to achieve better model performance especially classification accuracy on heterogeneous datasets. One of the ways to achieve this objective is to convert categorical attributes to numeric attributes. The one-hot-encoder scheme is one of the well-known methods for categorical to numeric transformation. However, known limitations are related to not knowing all the attributes' values in advance and the sparsity. Two models: First Come First Serve (FCFS) and High Distribution First (HDF) were introduced by One-hot Encoded Extended Compact (OHE-EC) technique. The objective of the technique is to extend OHE, by overcoming sparseness issue with compactness and without knowing distinct attributes in advance.

- **Objective 3 Enhancing Model Performance by Feature Engineering:**

  Feature engineering (FE) enables us to obtain additional information from current data through deriving new features. FE is one of the ways to improve an ML model's performance as the derived characteristics can assist in explaining the relationships in training data more precisely. Use of FE in FD is an understudied research area, but our studies have shown its' significance. There is a range of constraints to the current FE methods, which are either domain or context-oriented. One aspect is to increase the data dimension by applying the FE. The objective of the model is to improve the performance with FE using contextual expressions and external data. FE is applied with compact unified expressions (UE) with minimum prior knowledge of the domain of the dataset using profile models approach.

- **Objective 4 Ripple Down Rules based Fraud Detection Technique for Scalable Data:**

  As discussed earlier, most FD systems currently used by banks contain a rule-based component. However, rule-based systems can have limitations in terms of

adaptability, as well as being difficult to maintain over time. In comparison to the current rule-based techniques for FD, RDR is ideal, due to its less maintenance and incremental learning. Through prudence, RDR provides approach for better, faster rules additions and maintenance of the model. But the studies of RDR on distributed and Big data platforms are not done adequately due to lack of RDR tools on these platforms. The objective is to develop a single classification Unified Expression Ripple Down Rules-based fraud detection technique (UE-RDR) for scalable and distributed data. Three models: UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX have been developed and evaluated with the use of RDR. These are the compact minority, majority and mix of both class models. The developed algorithm is also implemented on Apache Spark platform.

1.3 Methodology Approach

FD techniques are based on ML, which require huge datasets for ongoing training for real-time monitoring. While banks provide information in certain cases, but usually the data is either in small quantities or it cannot provide particular characteristics required to validate new algorithms for FD. With these constraints in mind, the synthesized information generation is a feasible option. This research presents a framework for generating simulated online banking transaction data and assesses how well this simulated data correlates to the original, small reference dataset.

First of all, an innovative framework, Highly Correlated Rule Based Uniformly Distribution (HCRUD) was developed, which produces Synthetic Datasets for Experimental Validation of fraud analysis. The unique features of HCRUD are continuous attributes with predefined ranges, retaining attribute distributions in single and combination attributes, classification labels in data generated and large-scale data generation. Empirical findings are presented by comparing the data generated with the original reference data and by contrasting the distribution of the individual and the combination of the correlated attributes. Classification accuracy results are also observed with four well-known classification techniques (C4.5, RDR, Naïve Bayes and Random Forest). The empirical results show that the synthetic generated data retains features similar to the original reference data. This method can be used to generate synthetic data for any classification

domain; however, test data was created in this research to simulate bank transactions to analyze FD techniques in the banking domain. The datasets generated with this technique have been used in the subsequent research carried out to address the other objectives of this thesis.

ML algorithms provide improved efficiency on numeric data formats. However, bank transactions datasets are heterogeneous in nature. The objective is to achieve better model performance, especially better classification accuracy on heterogeneous datasets via categorical attributes conversion to numeric attributes. In this research, an innovative framework has been presented for categorical features transformation with compact One-hot encoder for FD in a distributed environment. We inferred a deficiency in OHE, introduced additional attributes based on contextual and model-based profiles and compressed sparse data further. This approach also incorporates two distribution and sorting based models. FCFS and HDF are two variants in the OHE-EC models. Classification accuracy, data size and efficiency evaluation for training and predictions models are carried out on Big data platform with the use of well-known classifiers including Random Forest, Decision Tree, Naïve Bayes, Support Vector Machine (SVM) and OneVsRest. In addition, an empirical assessment with a synthetic dataset generated from real bank transaction data and a well-known KDD-99 dataset has also been carried out.

FE is one of the ways to improve model performance. In FE research, a technique for FD by FE and compact UE technique is developed. Custom situated profile models (SPM) and ruleset compactness are used in this technique with minimal domain knowledge of the dataset. Empirical evaluation of the developed technique is performed with well-known classifiers (Decision Tree, RDR and Random Forest) using multiple datasets including bank datasets and two publicly available (German credit and Adult (Census Income)) datasets. Performance evaluation in terms of classification accuracy, precision, recall, f-measure, time and ruleset size is also done.

One of the challenges is to achieve higher accuracy for huge data, especially for the mixed dataset. To tackle this challenge, the thesis presents a Unified Expression Ripple Down

Rules-based FD Technique for Scalable Data. Empirical evaluation of the developed technique in terms of classification accuracy and ruleset compactness is performed with multiple datasets and compared with two of the RDR based RIDOR and Integrated Prudence Analysis (IPA) (Maruatona, 2013) classifiers. The evaluation is performed on bank reference, synthetic bank datasets and three publicly available datasets. The developed algorithm is also implemented on distributed and Big data ML platform, Spark. Subsection 1.3.1 highlights the details of implementation testbed on a Spark cloud environment.

1.3.1 Hadoop Experimental Setup

Spark (ASF, 2012) is a widely used open-source platform for large-scale data processing and very suitable for iterative ML tasks. It is much faster than conventional Hadoop (ASF, 2015) MapReduce. A multi-node Hadoop cluster with Spark was setup in Internet Commerce Security Laboratory (ICSL) on NECTAR research cloud (Moloney, Barker, Coddington, & Mecoles, 2011) to develop and evaluate the techniques for large datasets. The main parts in the cluster are spark gateway, history server, data nodes, node manager, name node and resource manager. CDHCluster551 is the Hadoop cluster version. Figure 1.1 shows a list of roles of different nodes in the Hadoop cluster.

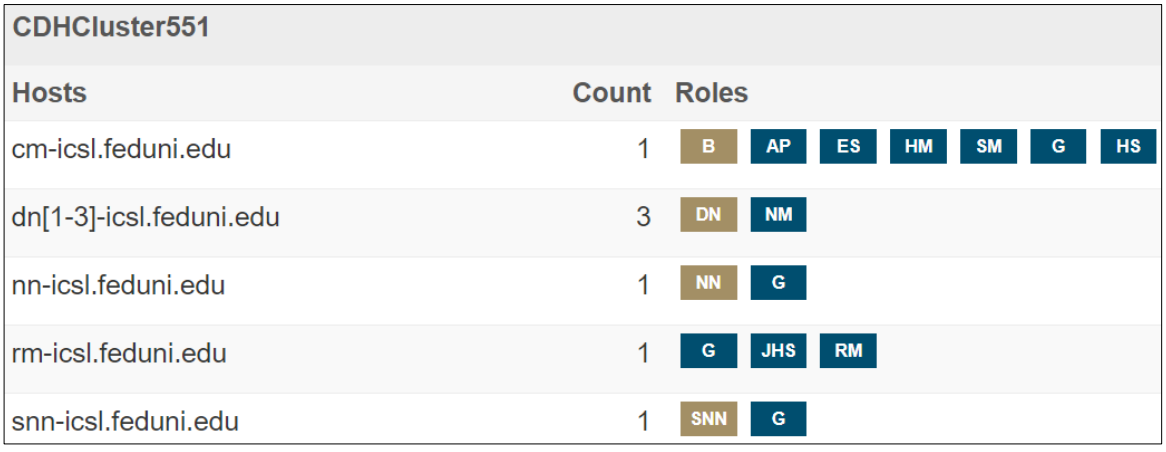


Figure 1.1: Hadoop and Spark Cluster Setup

1.4 Contributions and Publications

The thesis presents research on FD for online banking for scalable and distributed data. We aim to get a better understanding of the key structures of FD for online banking. Therefore, this thesis makes the following contributions in the area:

1.4.1 Development of a framework to Generate synthetic datasets for experimental validation of fraud detection.

Parts of this work has been published:

- Ul Haq et al. (Ul Haq et al., 2016)

  The 14th Australasian Data Mining Conference, Canberra, Australia

1.4.2 Development of a technique for Categorical features transformation with compact one-hot Encoder for fraud detection in distributed environment.

Parts of this work has been published:

- Ul Haq et al. (Ul Haq et al., 2018)

  The 16th Australasian Data Mining Conference, Bathurst, NSW, Australia

1.4.3 Development of a technique to Enhance model performance for fraud detection by feature engineering and compact unified expressions.

Parts of this work has been published:

- Ul Haq et al. (Ul Haq et al., 2019)

  19th International Conference on Algorithms and Architectures for Parallel Processing (Melbourne, Australia)

1.4.4 Development of a Single classification unified expression Ripple Down Rules fraud detection methodology for scalable and distributed data.

Parts of this work has been submitted for the publication:

- Ul Haq et al. (Ul Haq et al., 2020)

  AISC 2020 - Australasian Information Security Conference (Melbourne, Australia)

Figure 1.2 explains the overall research contributions in this thesis.

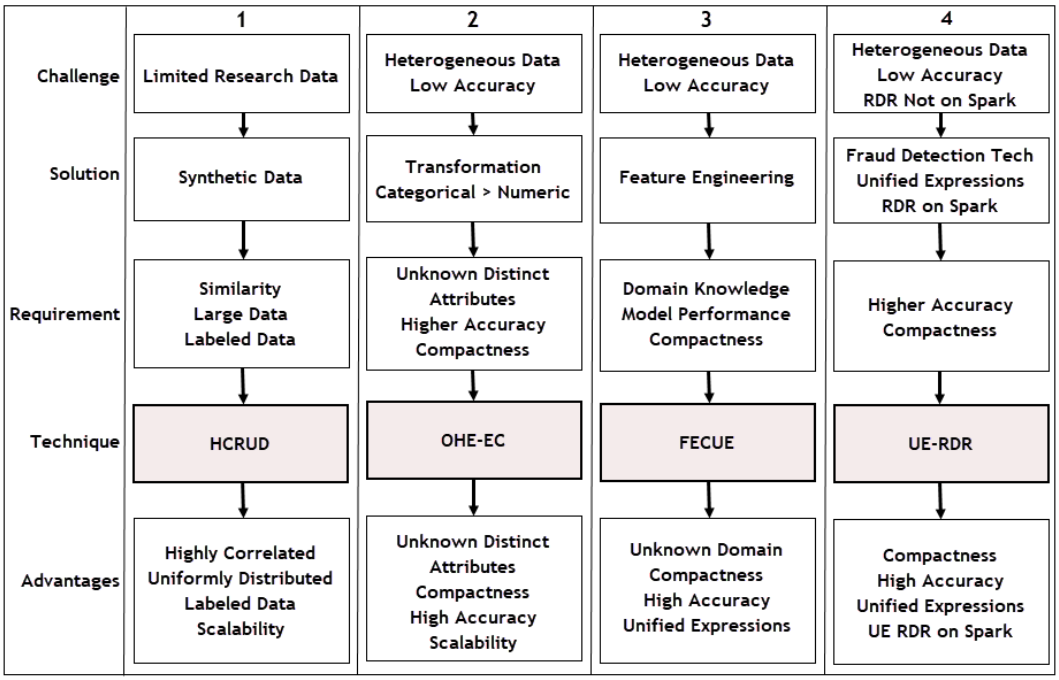


Figure 1.2: Schematic Diagram of the Overall Research

Figure 1.2 shows the diagram of the research as a whole. What were the challenges, solution and requirements, and how the challenges were overcome, which techniques were developed and what are the benefits of the developed techniques? The main challenges include the limited fraud analysis research data, conversion of categorical data to numeric data, model performance with FE and achieve higher accuracy for mixed and scalable data and implement the technique on Spark platform.

1.5 Structure of the Thesis

The thesis is organized into seven chapters, and contribution chapters of the thesis are based on four publications which are presented in chapters 3, 4, 5 and 6. Here is the outline of the thesis:

Chapter 1 introduces the thesis and presents the motivation for this research and the challenges. Then objectives of this research are presented, following subsections highlight the contributions and an overview of the structure of the thesis.

Chapter 2 presents an overview of existing FD systems, synthetic data generation, categorical feature transformation, model performance, FD, RDR, relevant data mining techniques and also highlights the research gaps in this area.

Chapter 3 presents a framework to generate synthetic datasets for experimental validation of FD. It describes the empirical evaluation of classification accuracy and correlation of generated data with the reference data. RMSE is also described as a performance metric for a root mean square error, in order to determine the difference between each data distribution and the combination of attributes in generated datasets compared to original reference datasets. This chapter is based on the publication: Ul Haq et al. (Ul Haq et al., 2016).

Chapter 4 presents an approach for categorical features transformation with compact One-hot encoder for FD in a distributed environment (OHE-EC). It describes FCFS and HDF models for OHE-EC technique. The chapter also presents an evaluation strategy to measure classification accuracy, the effect on data size and efficiency in terms of training and prediction model. Details of empirical evaluation of the proposed scheme with synthetic datasets generated from Bank's transaction data and the publicly available KDD-99 dataset are also presented. This chapter is based on the publication: Ul Haq et al.(Ul Haq et al., 2018).

Chapter 5 presents a technique to enhance model performance for FD by FE and compact unified expressions (FECUE). The chapter describes the use of custom and configurable SPM in the technique and in Ruleset compactness with minimum size dataset. Then explains the evaluation of the developed technique with multiple datasets and also describes evaluating metrics such as classification accuracy, precision, recall, f-measure, time and ruleset size. This chapter is based on a published paper: Ul Haq et al.(Ul Haq et al., 2019).

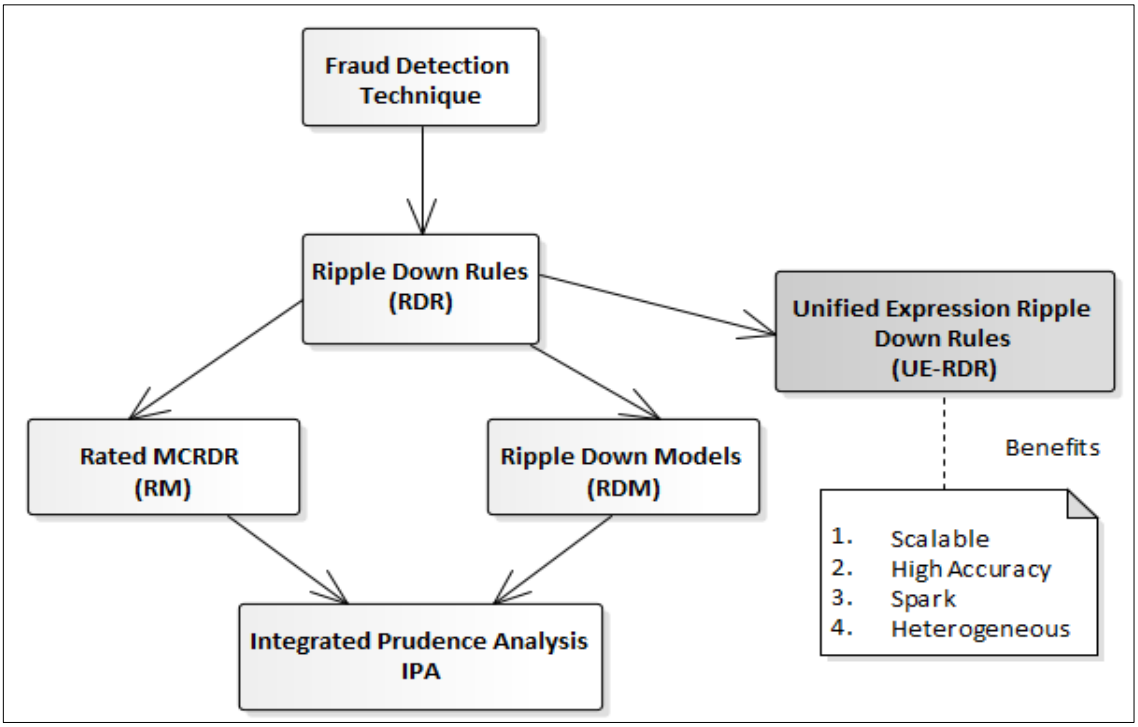
Chapter 6 presents UE-RDR technique. This chapter describes UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX models developed in the technique. It then describes the empirical evaluation of the developed classification accuracy technique and the compactness of the ruleset with multiple datasets and compares it with the RIDOR and IPA classifiers. The implementation detail of the technique developed on the distributed and Big data ML platform: Spark has also been presented. This work has been submitted for publication: Ul Haq et al.(Ul Haq et al., 2020).

Conclusions are presented in Chapter 7. This chapter highlights the importance and the challenges of FD research and then explains the challenges and the proposed solutions to these challenges. These challenges include limited research data, heterogeneous nature of the data and improving model performance in heterogeneous data with categorical data conversion to numeric data and with FE. Unified Expressions RDR based FD technique is also explained. Then, the chapter concludes the thesis, indicating some of the limitations and the possible future research work.

This chapter has provided introductory information on FD, the objective of the thesis, a list of publications as contributions. Synthetic data generation technique can help researchers to generate synthetic data from existing limited data or specific features to verify new research techniques and algorithms. Chapters 4, 5 and 6 of the thesis make use of the synthetic data to evaluate FD techniques with the use of scalable and distributed datasets. Chapter 2 provides a detailed literature review to establish the foundation of the research presented in chapters 3-6.

Chapter 2

Fraud Analysis Techniques



2.1 Introduction

Accurate FD in the banking industry can enhance the confidence of the clients in online banking. This thesis has focused on developing state of the art techniques with the use of artificial intelligence. This chapter presents the challenges faced in fraud analysis research, fundamentals of FD and techniques for FD, especially for online banking for scalable and distributed data. The significance of the research is highlighted by reviewing FD applications and techniques. In this chapter, a literature review is presented to highlight the need to develop a technique to produce synthetic datasets required for the experimental validation of fraud analysis. The thesis also argues that compact categorical features transformation technique using One-hot encoder can improve FD significantly. The thesis also shows that available features from the datasets might be not enough to detect fraud, so a unique FD improvement technique has been suggested with FE. Major banks normally have millions of customers and each customer could perform several transactions daily, which can result in billions of transactions daily. So a huge volume of data need to be processed. Incremental learning is one possible solution to the scalability problem in rule-based systems (RBS). In commercial FD systems, rules-based is a common approach. RDR tackles KBS issues related to KA and is suitable for reduced maintenance and incremental learning capabilities. Lower accuracy on heterogeneous data and lack of RDR implementation on the Spark large-scale data system are, however, some of the problems to be addressed in the RDR. Therefore, to address these real-world problems, this thesis also proposes a unified expression ripple down rules-based FD technique for large-scale heterogeneous data.

2.1.1 The Extent and Challenges of Online Banking Fraud

There is a big increase in different forms of frauds every year, resulting in substantial financial losses (Wei, Li, Cao, Ou, & Chen, 2013). As per Microsoft Computing Safety Index (MCSI) survey in 2014, the annual worldwide impact of phishing and various forms of identity theft could be as high as US$5 billion, while the cost of repairing damage to peoples' online reputation could be as high as US$6 billion, or an estimated average of US$632 per loss (Marican & Lim, 2014). PwC consulting mentions in the Global Economic Crime Survey 2016 that almost one-third of companies surveyed, reported to have been a victim of some kind of cybercrime. Also, the cybercrime was found to be the second most common types of economic crime analysed in the survey, while a recent survey reports

(Lavion, 2018) that there is 13% increase in fraud since 2016. IC3 is a valuable resource for victims of Internet crime and law enforcement agencies in identifying, investigating and prosecuting the crimes. (FBI, 2018) reports that IC3 received 14,408 complaints in 2018 which were related to technical support fraud from victims from 48 countries. The losses reported represents a 161% increase in losses from the previous year.

A global economic crime rate report by (PwC, 2016) indicates that financial services have proved to be the most endangered sector with an economic crime of 48% and the second most reported economic crime is cybercrime. Most commonly fraud domains include: Online banking, Credit Cards, Telecommunication, Healthcare Insurance, Online Insurance, computer intrusion, Online Auction (Abdallah, Maarof, & Zainal, 2016; Bolton & Hand, 2002; Carminati, Caron, Maggi, Epifani, & Zanero, 2015; John, Kennedy, Kennedy, Anele, & Olajide, 2016; Wei et al., 2013).

Figure 2.1 provides yearly statistics on reports by (FBI, 2018), which shows that crime allegations are rising every year and victim losses in 2018 were $2.71 Billion.

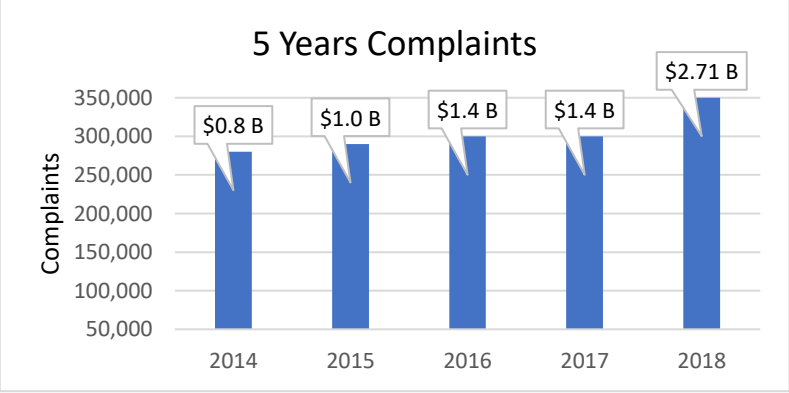


Figure 2.1: IC3 Last 5 Years Complaints (FBI, 2018)

Therefore, conducting FD research for online banking is a very important task, but a number of challenges in this area need to be overcome by research. Knowledge of banks' FD mechanism is very limited and banks do not publish statistics of the FD systems very often (Carminati et al., 2015; Maruatona, 2013). As most of the security is provided by

third-party IT companies who also protect the intellectual property from their competitors. So both banks and security IT companies do not publicise most of the information on their security systems.

(Bolton & Hand, 2002) also highlight that development of new FD methods is difficult because the exchange of ideas in FD is very limited, but authors also support the notion that FD techniques should not be described publically with details; otherwise criminals may also access that information (Carminati et al., 2015). (Phua et al., 2010) highlight that FD using data mining techniques is very common in the industry.

2.1.2 Security in Online Banking

Online banking systems (OBS) have different security mechanisms to prevent unauthorised access to fraudsters. Despite all these advances, unsuspecting victims still lose their credentials to phishing fraudsters and online identity thieves. When a fraudster gains access to a user's bank account, it is not easy to detect that activity. Different banks are using different FD systems. Some of the well-known commercially used FD systems for online banking are FICO, PRM, Kount and SAS Fraud Manager.

2.1.2.1 Commonly Used Commercial Fraud Detection Systems

The FICO system is one of the most widely used FD systems by banks globally (Brabazon et al., 2010; FICO, 2010); it uses a neural network and a rule-based engine. It has real-time capabilities. FICO is used by financial institutions, banks, manufacturing and credit unions, primarily for FD of debit, credit, deposit and ePayments (Capterra, 2019). Another widely used FD system is PRM (ACI, 2011). Like FICO, it also uses a neural network and a rules-based approach. The PRM system is used in over 40 countries including eight of the top 20 banks in the world. A variety of debit & credit card fraud, account fraud and money laundering fraud may be identified by PRM (Hafiz et al., 2016). One such system for FD and investigation system is SAS Fraud Manager (SAS, 2007), which is mainly FD solution for debit and credit cards; and has the real-time capability. This FD system uses anomaly detection (AD) technique. Another fraud prevention system is (Kount, 2006), which uses both supervised and unsupervised ML models. Some of the world's largest payment services providers, gateways, wallets and processors use this system. Few other FD and preventions systems are (FraudNet, 1997; PatternSpy, 2015; RiskNet, 1998). (Hafiz et al.,

2016) research carries out a detailed comparison of most of these commercial vendors. However, a rules-based approach is the common thing in all these commercially used FD systems (Capterra, 2019).

2.1.2.2 Shortcomings in Commercial Fraud Detection Systems

The above mentioned commercial fraud detection systems are widely used by the banks and finance institutions; however, no FD system is 100% effective. (Dehaven, 2014; Hafiz et al., 2016; Herschel, Linden, & Kart, 2015) have highlighted some of the shortcomings in commercial fraud detection systems.

- These systems lack scoring and consumer location tracking for the mobile device transaction.
- Integration layer for these systems for importing data is not perfect.
- Not all systems have the capability for logging data encryption.
- Systems consider larger amounts of transactions and mostly disregard smaller amounts.

2.1.3 Rule-Based Systems (RBS)

Rule-based systems are part of a large group of approaches that attempt to model a human‘s expertise called Knowledge-Based Systems (KBS). Traditional KBS including RBS have common issues of KA, brittleness and incremental learning. The term brittleness is coined by (Lenat, 2006). It refers to software that is likely to come to an incorrect result when faced with some unexpected scenario. Whereas incremental learning is one of the possible solutions to the problem of scalability, where data is processed in parts and then combines the result to reduce the memory use (Syed, Huan, Kah, & Sung, 1999).

2.1.4 Prior Work on Fraud Detection

Online fraud, internet fraud and cybercrime are broad-based and occur in many ways. Online fraud can be described as any illegal act committed online. Internet banking fraud, Mobile banking, Phishing, Mule recruitment, Shopping and auction site fraud, Scams, Spam, and Identity theft (AFP, 2015) are various types of online frauds. Internet banking fraud is a fraud that is undertaken using any online technology to illegally remove money from or move it to another bank account. The literature review is divided into multiple areas

including synthetic data generation, categorical data transformation to numeric, FE and FD to address the research objectives of the thesis.

2.1.4.1 Intrusion Detection Systems (IDS)

According to (Lee, Park, Eom, & Chung, 2011), Intrusion Detection Systems (IDS) can provide a greater amount of safety, but in ratio to safety intensity, it requires much more computing resources. The authors proposed a multi-level IDS and log management for effective IDS in cloud computing. It contributes to the efficient use of resources by introducing a significant amount of safety responsibility to customers depending on the type of anomaly method, which connects users according to anomaly rate to distinct safety organizations. (C.-M. Chen, Guan, Huang, & Ou, 2012) believe that hackers can conduct a series of assaults on a secure destination system in the cloud, for instance, by evading a cloud-based easy-to-use device and then using the prior backdoor to attack the system. The suggested detection system analyses various logs from the cloud to obtain the meanings of log activities. For the small amount of offences, suspicious activities are often overlooked by the administrator. Hidden Markov Model is implemented to model the sequence of attacks performed by the hackers and such stealthy occurrences over a long-time frame as it will become important in the state-aware model. The systems suggested by (C.-M. Chen et al., 2012; Lee et al., 2011) for IDS, focus mainly on detecting potential events, recording data and monitoring efforts.

2.1.4.2 Anomaly Detection (AD)

(Ilgun, Kemmerer, & Porras, 1995) propose a rule-based intrusion detection (ID) technique and acknowledge that AD is one of the earliest approaches to the ID and rule-based methods of AD are implemented in recent years. (Chiu, Yeh, & Lee, 2013) discuss that the hijacking of the account or service in cloud computing is more dangerous. The authors suggested a framework with AD technique to profile a user's ordinary behaviours. When a user profile is discovered from the information gathered, the alerts will be triggered by all suspect behaviours identified by the profile. The alerts will be sent to the database and the account holder and the cloud manager will be notified. There are two main components of the framework. The first portion is the data collector and the second portion is the learning module, which has introduced various ML techniques to mine the frequent trends from training data and obtain a suspect rating limit. If the suspect rating exceeds the established limit, the transaction will be reported as an anomaly by the scheme. (Chiu et

al., 2013) suggest that AD technique sends warning to system users and is more cloud-focused and not appropriate for large information. (Brabazon et al., 2010) also describes the use of an AIS-based approach to AD to reduce credit card fraud.

Abnormality analysis for streamed log data is performed by (Harutyunyan, Poghosyan, Grigoryan, & Marvasti, 2014). The authors expand the notion of time series data strategic thresholding to any type of information that is a stream of documents and activities. They implemented the concept of the normalcy of those streams in their suggested method and developed a mechanism for their abnormality detection in run-time flow. Under unique limitations on complexity and scalability, they implemented a fresh decision-making structure for retrieving information from data flows. The technique proposed by (Harutyunyan et al., 2014) is primarily to analyze abnormality of event data from log files and to obtain useful information from source and types of events.

2.1.4.3 Fraud Detection Techniques and Approaches

Many ML methods to combat fraud have been developed. Common FD techniques include ANN, ES (Knowledge-Based (KB)), Inference Engine and Data Mining (Maruatona, 2013; Wei et al., 2013). Most widely used approaches for FD are supervised, unsupervised, semi-supervised and hybrid methods (Abdallah et al., 2016; Bolton & Hand, 2002; Chandola, Banerjee, & Kumar, 2009; Hodge & Austin, 2004; John et al., 2016; Wei et al., 2013).

(Kovach & Ruggiero, 2011) also suggest an FD system for online banking centred on the local and global analysis of users' behaviour. Differential assessment is used to obtain evidence of fraud where a substantial variation from normal behaviour reveals a potential fraud. Fraud evidence is focused on the number of user accesses and a probability value that differs over the period. Their suggested technique of FD focuses on efficient identification of devices used to control accounts and evaluate the likelihood of being a fraud by monitoring the number of distinct records that each device accesses.

A method for detecting credit card fraud is furnished by (Duman & Ozcelik, 2011). The authors proposed a mixture of the two well-known meta-heuristic methods for ranking, which are: genetic algorithms and scatter search. The technique is implemented to the actual

data and the findings achieved are very effective relative to the present system in use. Each transaction is rated with this approach and the operations are categorized as fraudulent or legitimate depending on the results. They believe that an FD scheme is better than the scheme that detects many low-risk frauds, which detects crime even less in amount but greater in value.

(Kou et al., 2004) also carry out a Data Mining-based study of FD research to categorize the research on four primary methods including supervised, hybrid, semi-supervised and unsupervised approaches and also identified the relationship of FD with other fields. An FD strategy is suggested by (Herland, Khoshgoftaar, & Bauder, 2018) for Medicare fraud using three medicare and medicaid services datasets. Their method operates on the mixed dataset by connecting several training datasets. The authors used three classifiers: Random Forest, Gradient Tree Boosting and Logistic Regression models and used the area under the Curve (ROC) metric to measure FD performance. They concluded that the highest output in FD is on the combined dataset. Dataset size is not discussed, but this technique is not optimal for big datasets where another dataset with a mixture of initial datasets is required.

(Bai, 2013) proposes a method which is primarily an effective search option for real-time information, but is not appropriate for the classification domain, where each operation must be categorized as fraud or not a fraud. (Kovach & Ruggiero, 2011) suggest fraud tracking system is focused on the behaviour of the customers and depends heavily on device access, which is not appropriate for large, real-time information. Credit card FD solution proposed by (Duman & Ozcelik, 2011) uses meta-heuristic approaches. However, the authors did not provide methods for dealing with large information and information in real-time.

(Wei et al., 2013) present an FD technique for highly imbalanced data using a ContrastMiner algorithm, which mines contrast patterns and differentiates fraud from genuine behaviour. (Kou et al., 2004) believe that FD research mainly utilizes data mining, statistics, and artificial intelligence and fraud is recognized from anomalies in data and patterns.

A proactive risk identification is suggested by (Khorshed, Shawkat Ali, & Wasimi, 2012). In their threat identification model, ML methods (including rule-based learning and statistical learning theory) and a large repository of threats are used. Comparison of ML techniques, including Naïve Bayes, Multilayer Perceptron, SVM, Decision Tree, and PART, is made to rank into attack category. Their suggested threat detection model also issues alerts to the users involved, but it is not appropriate for large data as their model has not addressed scalability issues.

2.1.4.4 Deep Learning

Deep learning (DL) is a subset of machine learning in which ANN learns from large quantities of data (Goodfellow et al., 2016; Heaton et al., 2016; LeCun, Bengio, & Hinton, 2015). Conventional ML techniques were unable to process data in raw form. In order to extract features, ML requires careful engineering and domain knowledge to do feature extraction from raw data, on the other hand DL needs no engineering as it automatically extracts features from raw and historic data (Roy et al., 2018). DL is an emerging research area and has achieved great success in many machine learning domains, including fraud detection (Chalapathy & Chawla, 2019; Chauhan & Singh, 2018; El Bouchti, Chakroun, Abbar, & Okar, 2017; Kazemi & Zarrabi, 2017; Q. Zhang, Yang, Chen, & Li, 2018). The most common DL models include auto-encoder (SAE), deep belief network (DBN), convolutional neural network (CNN) and recurrent neural network (RNN). Long short-term memory (LSTM) is a variant of RNN. Decision trees, Random forest, SVM, and Naïve Bayes are shallow machine learning methods, which require feature extraction.

Although DL is its own benefits, however millions of data is required to train an efficient training model, needs high computational power (Chauhan & Singh, 2018; Heaton et al., 2016), while in ML model can also be trained with small data set. (W. Chen, 2016) indicates that ML models are memory consume high memory and need industrial sized clusters or high-performance graphics processing units. (Altexsoft, 2017) highlights that the lack of interpretability is a major problem with deep neural networks, so it is practically impossible to define how the system came to one or the other conclusion.

2.1.5 Outlier Detection (OD)

An outlier is more precisely defined by (Hawkins, 1980) as “an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism”. The Outlier has also been identified by (Bakar, Mohemad, Ahmad, & Deris, 2006) as data point which is very different from the rest of the data based on some measure. OD has been used for centuries to detect and where appropriate, remove anomalous observations from the data (Hodge & Austin, 2004). Fraudulent transaction in banking can be considered outliers, and OD is an integral part of the FD. Outliers can be human or instrument errors, natural deviations in populations, fraudulent behaviours, changes in behaviour or faults in systems (Hodge & Austin, 2004). Univariate and Multivariate methods are two main approaches in the literature for OD, whereas Parametric and non-parametric are the main categories of the OD techniques (Ben-Gal, 2005). The author further explains that non-parametric are model-free methods, while that parametric are statistical methods.

2.1.5.1 Applications of Outlier Detection

(Dokas et al., 2002) have mentioned some of the use of OD in area of credit card FD, the discovery of criminal activities and ID. However, (Bakar et al., 2006; Hodge & Austin, 2004) have given a comprehensive list of the applications of OD: FD, loan application processing, ID, activity monitoring, network performance, fault diagnosis, structural defect detection, satellite image analysis, detecting novelties in images, motion segmentation, time-series monitoring, medical condition monitoring, pharmaceutical research, detecting novelty in text, detecting unexpected entries in databases, detecting mislabelled data in a training dataset. The authors conclude that analysing OD is an interesting and important activity in data mining.

2.1.5.2 Outlier Detection in Network Intrusion Detection

The ID is unauthorised access in computer networks. ID involves recognizing a number of malicious behaviours that compromise information's integrity, confidentiality and accessibility (Dokas et al., 2002). There are different approaches in ID and prevention systems, which are host-based, network-based and application-based (Liao, Lin, Lin, & Tung, 2013; Patel, Qassim, & Wills, 2010). Host-based systems protect servers and workstations, while network-based systems protect network segments. Two categories of ID techniques are AD and signature detection (SD) or misuse detection (MD) (Dokas et al.,

2002; Patel, Taghavi, Bakhtiyari, & Celestino Júnior, 2013). FlowMatrix and SNORT are examples of AD and SD or MD respectively. (Maruatona, 2013) has highlighted various challenges in IDS.

- Process/manage large data volume
- Detect as much anomalous behaviour as possible
- Have real-time detection capabilities
- Still not fully reliable
- Not able to detect novel patterns
- Need to adapt intelligent programming techniques and KBS to improve detection rates

2.1.5.3 Outlier Detection in Fraud Detection

FD system uses AD or OD method. FD is about identifying and recognizing malicious operations or criminal activities by the schemes and reporting them to a machine manager (Behdad, Barone, Bennamoun, & French, 2012; Chandola et al., 2009).

2.2 Synthetic Data Generation

To conduct research on Fraud analysis, bank data is required. Sometimes banks provide data, but the data given by the bank is either in small volume or it may not meet specific features which are needed to verify new research techniques and algorithms. With the consideration of these limitations, a viable alternative is to generate synthesized data.

The idea of synthetic data generation is not new, as (Rubin, 1993) generates data to synthesize the Decennial Census long form responses from the short form households using multiple imputations. However, it has not previously been applied to the area of FD for online banking. Synthetic data can be used in several domains; benefits of synthetic data are well presented by (Bergmann, 2015).

- It allows controlling the data distributions used for testing. So, the behaviour of the algorithms under different conditions can be studied.
- It can help in performance comparison among different algorithms. For example, for evaluating the scalability of the algorithms.

- It creates instances having the finest level of granularity in each attribute. But in publicly available datasets anonymization procedures is applied due to privacy constraints.

Various attempts have been made to generate synthetic datasets; one such technique uses uni-modal cluster interpolation, e.g. Singular value decomposition interpolation (SVDI) (Coyle, Roberts, Collins Jr., & Barbu, 2013). This technique presents a method that uses data clusters at certain operating conditions where data is collected to estimate the data clusters at other operating conditions, thus enabling classification. SVDI's main shortcoming is that the estimates of data clusters and known data clusters all have the same number of samples.

Different frameworks to synthesise the data (Bergmann, 2015; Keen, 2015; Maj, 2003; Wisser, 2015) have been studied, but all of these frameworks neither classify the data nor are based on any existing datasets. (Jeske et al., 2005; P. J. Lin, Samadi, & Jeske, 2006) suggest synthetic datasets generation techniques, but their techniques are based on complex semantic graphs and support the testing and training of discovery and analysis systems. One attempt to generate synthetic census-based micro-data is with the customization and using extensibility of an open-source Java-based system (Ayala-Rivera, McDonagh, Cerqueus, & Murphy, 2013). In the data generation process, authors use probability (Haigh, 2013; Tijms, 2012) weights by capturing frequency distributions of multiple attributes. Due to attribute interdependency, they also apply attributes constraints, but they have not applied the weight on the combination of attributes. It might be possible that distribution on individual attributes is same in generated data, but this distribution might be different if checked on the combination of attributes. The generated data cannot be used in the domain of classification problems, as this is not the classified data. Another attempt is constraint-based automatic test data generation. The technique is based on mutation analysis and creates test data that approximate relative adequacy (Demilli & Offutt, 1991). But this technique is only used to generate test data for unit and module testing. (Christen & Pudjijono, 2009) also propose a synthetic data generation technique, however the technique is limited to the personal information attributes and more specific to individuals, households and families.

(Chawla, Bowyer, Hall, & Kegelmeyer, 2002) present synthetic minority over-sampling-based construction of classifiers from imbalanced datasets. Under-sampling of the majority (normal) class has been proposed as a good means of increasing the sensitivity of a classifier to the minority class. Their method of over-sampling the minority class involves creating synthetic minority class examples (Chawla et al., 2002). This approach is ideal in imbalanced data scenario where the requirement is to reduce the majority class and increase the minority class. This technique is not ideal for increasing overall data size.

(Yoo & Harman, 2012) have suggested a technique to generate additional test data from existing reference data. Their paper highlights that mostly existing automated test data generation techniques tend to start from scratch, implicitly assuming that no pre-existing test data are available. The authors suggested that pre-existing test cases could be used to assist the automated generation of additional test cases. The authors have used a search-based test data regeneration technique, that can generate additional test data from existing test data using a meta-heuristic search algorithm. But the generated data, cannot be used in the domain of classification problems, as it does not have classification labels.

2.2.1 Classification Techniques Used for Data Validation

The system can be trained with generated datasets and tested on bank dataset. Classification accuracy of the generated dataset can be observed and compared with four well-known classification techniques, which are Decision Tree (Quinlan, 1992), RDR (Compton & Jansen, 1988; Richards, 2009)**,** Naïve Bayes (Swain & Sarangi, 2013) and Random Forest (Breiman, 2001).

2.2.2 Instance-Based Learning (IBL)

(Aha, Kibler, & Albert, 1991) have presented an instance-based learning (IBL) framework, which generates classification predictions using only specific instances by applying similarity functions. IB1 and IBk are instance-based learners (IBL) (Chilo, Horvath, Lindblad, & Olsson, 2009) which can be used for testing the classification accuracy. IB1 is the simplest IBL, nearest neighbour algorithm where similarity function is used. It classifies the instance according to the nearest neighbour identified by Euclidean distance approach

(Aha et al., 1991; Chilo et al., 2009). IBk is similar to IB1, but the difference is that in IBk, the K-nearest neighbours are used instead of only one. Three different distance approaches are employed in IBk, including Euclidean, Chebyshev and Manhattan Distance (Chilo et al., 2009).

2.3 Categorical Features Transformation

FD for online banking is an important research area, but one of the challenges is the heterogeneous nature of transactions data, i.e. a combination of numeric as well as mixed attributes. Numeric type information generally provides a better ranking, regression and statistics clustering efficiency. In an FD research for online banking by (Maruatona, 2013) shows that numeric datasets give better accuracy as compared to categorical or mixed datasets. (Z. Huang, 1998) points to the well-known efficiency of the k-means algorithm in the clustering of large data sets. But, the algorithm only operates on numerical data. However, often real-life data mining problems do not only have numerical or categorical characteristics. In addition, some ML platforms such as Apache Spark accept numeric data only. OHE is a widely used approach for transforming categorical features into numerical features in traditional data mining tasks. The One-hot approach has some challenges as well: the sparseness of the transformed data and the distinct values of an attribute are not always known in advance. Model accuracy and compactness of ML models are equally important due to growing memory and storage needs.

Several efforts have been made in the past to transform categorical attribute to numeric attributes. First attempt to convert a categorical feature to a numerical is OHE, but this transformation results in high-dimensional sparse-data. (Jian, Cao, Pang, Lu, & Gao, 2017) have transformed categorical data with Coupled Data Embedding (CDE) technique by extending coupling learning methodology by obtaining hierarchical value-to-value cluster couplings. CDE is slower than other embedding methods, thus is not ideal for large datasets. It is only applied to unsupervised clustering domain. Another categorical data-representation technique is proposed by (Qian, Li, Liang, Liu, & Dang, 2016) with an objective of solving the problem of the categorical data not having a clear space structure. The authors have not addressed the problem of clustering for the mixed dataset. A comparative evaluation of similarity measures for categorical data is done by (Boriah,

Chandola, & Kumar, 2008). But the evaluation is performed in a specific context of OD, and relative performance of similarity measures is not studied for classification and clustering. The authors highlight that several books on cluster analysis (Anderberg, 1973; Hartigan, 1975; Jain & Dubes, 1988) that discuss the problem of determining the similarity between categorical attributes, recommend binary transformation of data for similarity measures. One of the suggested technique to convert categorical attributes to numeric attributes for large datasets is by (Hsu, Chang, & Lin, 2010), while another approach is proposed by (Z. Huang, 1997). Similarly, (Shih et al., 2010) also propose a technique for clustering mixed categorical and numeric data using a two-step method with TMCM algorithm. But the assumption with these techniques is that all attributes values must be known before constructing their new value, which is not applicable in real-time data.

(Cha, 2007) states that there are a considerable amount of distance/similarity tests in many different areas. The author also indicates that the shortest distance between two points is a line, originally stated by Euclid. One of the suggested techniques is the use of resemblance characteristics of the attributes and then the use of an appropriate variable to transform this resemblance to numerical form. A broad range of distance functions and similarity estimates, including Euclidean distance, cosine similarity, and relative entropy, have been used in the ML algorithms (A. Huang, 2008). The Euclidean distance between vectors X and Y is defined as the square root of the sum of squared differences between corresponding elements of the two vectors.

2.3.1 Distributed and Parallel Data Processing Platforms

With the availability of inexpensive computing and processing resources, companies store a lot more information to extract knowledge with the use of Big data analytics. Online banking transaction records are also continually growing. The volume of information is becoming very big and traditional methods of handling and processing big information are no longer working. It is commonly recognized that we are facing an age of data explosion. To process and store large data, we need platforms which support distributed and parallel data processing. (Kambatla, Kollias, Kumar, & Grama, 2014) acknowledge today's widespread recognition of applications involving efficient analyses of large datasets, and enhanced software solutions must take into account large datasets.

2.3.1.1 Apache Hadoop

Apache Hadoop (ASF, 2015; White, 2015) is an open-source implementation of MapReduce and a framework for distributed storage and processing of huge datasets on computer clusters built from commodity hardware. A typical MapReduce program is composed of two phases: a map and a reduce phase. Map phase processes the input, while reduce performs a summary operation. A typical Hadoop cluster consists of a name node, a resource manager and multiple data and worker nodes.

2.3.1.2 Machine Learning with Hadoop

Apache Spark is a Big data Processing analytics platform with built-in ML modules (Dean & Ghemawat, 2008). (Pentreath, 2015) indicates that Apache Spark is optimized for low-latency tasks and to store intermediate data and results in memory, to address some of the major drawbacks of the Hadoop framework. Pentreath further says that the design of ML models is typically iterative, so Spark is suitable for this case of use.

2.3.1.3 Hadoop and Spark for Machine Learning

Various techniques are developed on Hadoop to solve ML problems of different domains. ScalParC technique is used (Joshi, Karypis, & Kumar, 1998) for classifying large datasets. The authors have used a parallel formulation of Decision Tree-based classification process. But this technique is not implemented in FD domain yet. MapReduce (Dean & Ghemawat, 2008) has proved to be an effective method of dealing with big datasets in these circumstances. (Khan, Shakil, & Alam, 2015) believe that MapReduce is one of the most common data processing models on computer clusters. However, (X. LIN, WANG, & WU, 2013) argue that Hadoop MapReduce is not suitable for the applications which reuse a dataset across multiple parallel operations, which include further iterative ML algorithms, as well as interactive data analysis tools. The authors proposed that the Hadoop platform should be incorporated with Apache Spark (Ryza, Laserson, Owen, & Wills, 2015) to facilitate such applications and effective in-memory computations. Advantages of Spark over conventional MapReduce are well explained by (X. LIN et al., 2013):

- Spark is a memory-based framework and is suitable for iterative algorithms and interactive ad-hoc queries.

- Spark supports a Directed Acyclic Graph (DAG) type schedule instead of the only Map and Reduce phase. It avoids materializing the intermediate records through pipeline operations to decrease I/O operations.
- Task scheduling is with low latency in the Spark system. It uses an event-driven architecture and can launch tasks in just 5ms.

Use of the Hadoop MapReduce method (Map and Reduce stages) (Vernekar & Buchade, 2013) introduces a concept for large volume log file analysis in a distributed environment. The authors state that log files are commonly used for the purpose of security threat identification. These problems and threats are identified by detecting the suspicious pattern of events in the log file. Since the server log files are very big in volume, handling such a big log file involves both adequate approach and resources. The large log file is divided into blocks and presented as an input to the Map stage. These chunks are then allocated to several Map tasks located on the Hadoop cluster servers, enabling parallel processing of different data parts and generating quicker performance of the marginal key-value pair. Security threat identification technique suggested by (Vernekar & Buchade, 2013) uses the Hadoop MapReduce methods that can be used for large data, but they are not appropriate for real-time information, in-memory handling and iterative processes. MapReduce is a batch processing model in which the model should be periodically re-trained. However, (Pentreath, 2015) believes that using this strategy to update models is not viable as fresh information comes instantly.

(Bai, 2013) proposes a technique for searching Big data from log files in real-time. Bai suggests that companies could mine business value, particularly if it can be accomplished in real-time. But there is a challenge to handle big log files because traditional technology is not strong enough to handle enormous information. Hadoop echo system offers a new way for Big data processing. ElasticSearch is an open-source and modern search engine for the cloud environment. Bai suggests a Big data query technique, which is centred on contemporary distributed systems and ElasticSearch cloud-based real-time search tool.

(Hayes & Capretz, 2014) propose a contextual AD technique for streaming sensor networks. Authors recommend that predictive modelling, such as detecting anomalies, is a major challenge in large data. As more and more Big data streams are produced from natural detectors, logging apps, and the Internet of Things; this issue becomes more complicated. Furthermore, most current AD methods only recognize the information itself, regardless of the information background. As information becomes more complicated, bias tracking methods for the background are becoming increasingly essential. The authors' suggested research describes a contextual method for detecting anomalies for use in streaming device networks. For real-time place AD, the method utilizes a well-defined information AD method. In addition, a post-processing contextual AD algorithm is provided depending on sensor models produced by a multivariate clustering algorithm, which are sets of contextually comparable detectors. (Melo-Acosta, Duitama-Munoz, & Arias-Londono, 2017) also introduce a credit card fraud identification method in Big data framework, but their method is more applicable to imbalanced and unlabelled data.

2.4 Feature Engineering

The performance of ML models can be improved in a variety of ways including segmentation, treating missing and outlier values, FE, feature selection, multiple algorithms, algorithm tuning/compactness and ensemble methods. FE and compactness of the model can have a significant impact on the algorithm's performance but usually requires detailed domain knowledge. Accuracy and compactness of ML models are equally important for optimal memory and storage needs. Literature focuses on FE and compactness of rulesets. The compactness of the ruleset can make the algorithm more efficient and derivation of new features makes the dataset high-dimensional potentially resulting in higher accuracy.

Some of the known methods of improving model performance are segmentation (Bijak & Thomas, 2012), treating missing (Xiaofeng, Shichao, Zhi, Zili, & Zhuoming, 2011) and outlier values, FE (Turner, Fuggetta, Lavazza, & Wolf, 1999; Xu, Hong, Tsujii, & Chang, 2012; Yu et al., 2010)**,** Feature selection, Multiple algorithms and Algorithm tuning. Segmentation divides the population into several groups. FE is about extracting more information from existing features. Feature selection is finding the most important subset

of features. Multiple algorithms are the application of the relevant model to see better suitability of models for a particular domain. Algorithm tuning is the optimum parameter values used in a particular ML algorithm.

Our research focuses on FE, which is being used in different domains to improve model performance. In (Yu et al., 2010) authors have conducted an educational data mining study; and evaluated FE for KDD Cup 2010 by training the model from students' past behaviour and then predicting future performance. Authors in (Xu et al., 2012) have designed an information extraction technique using FE with a combination of rule-based and ML methods. This technique is applied in the medical domain for narrative clinical discharge summaries. In another research (Turner et al., 1999) have proposed the concepts of FE and evaluating the impact of FE on the software development life cycle. The authors proposed their research as the first step towards the development of FE and its relationship to other domains. A text classification FE technique is developed by (Garla & Brandt, 2012), which is guided by the ontology. This technique utilizes the domain knowledge encoded in the taxonomical structure of the medical language system with the help of context-dependent relatedness between pairs of concepts. This technique is developed for clinical text classification in the medical domain. (Bahnsen, Aouada, Stojanovic, & Ottersten, 2016) suggest an FE strategy to create a new range of features based on an analysis of the normal transaction actions using the von Mises distribution. The methodology of the authors is primarily for credit card FD and they evaluated normal transaction time behaviour, using transaction aggregation strategy. However, this approach is not ideal for large datasets.

These developed techniques have a variety of limitations and are either domain or context-specific. The authors do not discuss the problem or the solution related to the need to increase the data dimension with the use of FE techniques. Also, the performance in terms of classification accuracy, time and model's size is not discussed. FE via external sources is also not used in these techniques.

2.5 RDR Based Fraud Detection Technique for Scalable Data

2.5.1 Ripple Down Rules (RDR)

(Compton & Jansen, 1988) propose RDR to tackle maintenance and KA issues in KBS. RDR is an approach to KA. RDR has significant advantages over conventional rule-bases; including.

- Better, quicker and less costly rule addition and maintenance methods.
- Prudence in RDR systems enables the model to realise when a current case goes beyond the competence of the (Notify admins to investigate the situation).

(Littin, 1996) describes RDR as a binary tree like construct where each node matches to a rule and that the binary tree is identical to CART and ID3 (Breiman, Friedman, Olshen, & Stone, 2017; Quinlan, 1986). The author also highlights that inclusion of RDR top-level empty rule is used generally with a default class. The author further explains that the root node, is always true by default, and is connected to a network of nodes, and 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.

Figure 2.2 demonstrates that the RDR structure is a binary tree type. Horizontal solid lines are the true branches, whereas false branches are represented by vertical dashed lines. The grey shaded boxes represent evaluated nodes. From these shaded nodes, the nodes which are evaluated to be true are represented by bolded boxes.

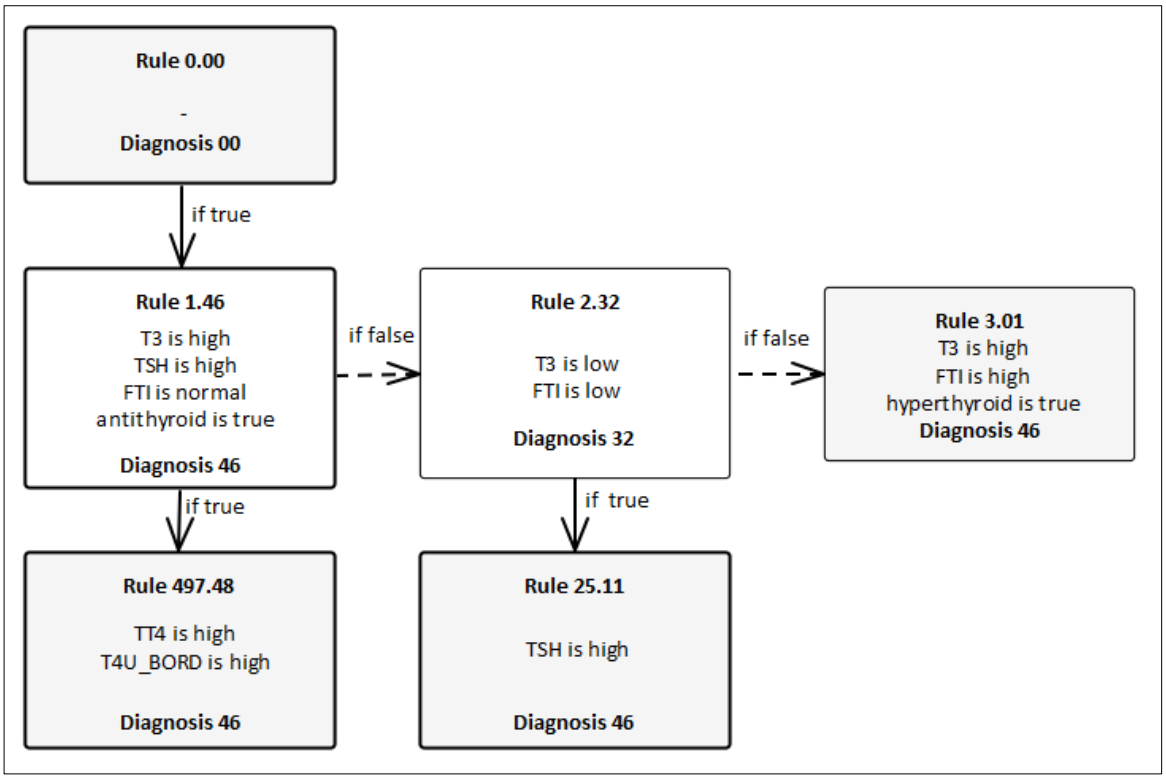


Figure 2.2: RDR Tree Structure (Gaines & Compton, 1995)

Later on (Kang, Compton, & Preston, 1995; Yang, Sung, Edward, & Byeong, 2004) implement multiple classifications RDR (MCRDR). MCRDR is developed due to the lack of single classification RDR to handle multiple diagnoses, for example, multiple conclusions in cases where patients have more than one disease, i.e. to handle more than one classification cases. Inference procedure in MCRDR can be described using an instance situation where the root node is first assessed and then all root nodes are subsequently tested. The nodes that are evaluated to be accurate will test their children's nodes and the rippling method proceeds until the last node is reached. Figure 2.3 explains the inferencing process in an MCRDR structure. In the diagram, an example case is taken where X= {b,d,f,k,o,h,e,m,t,y}.

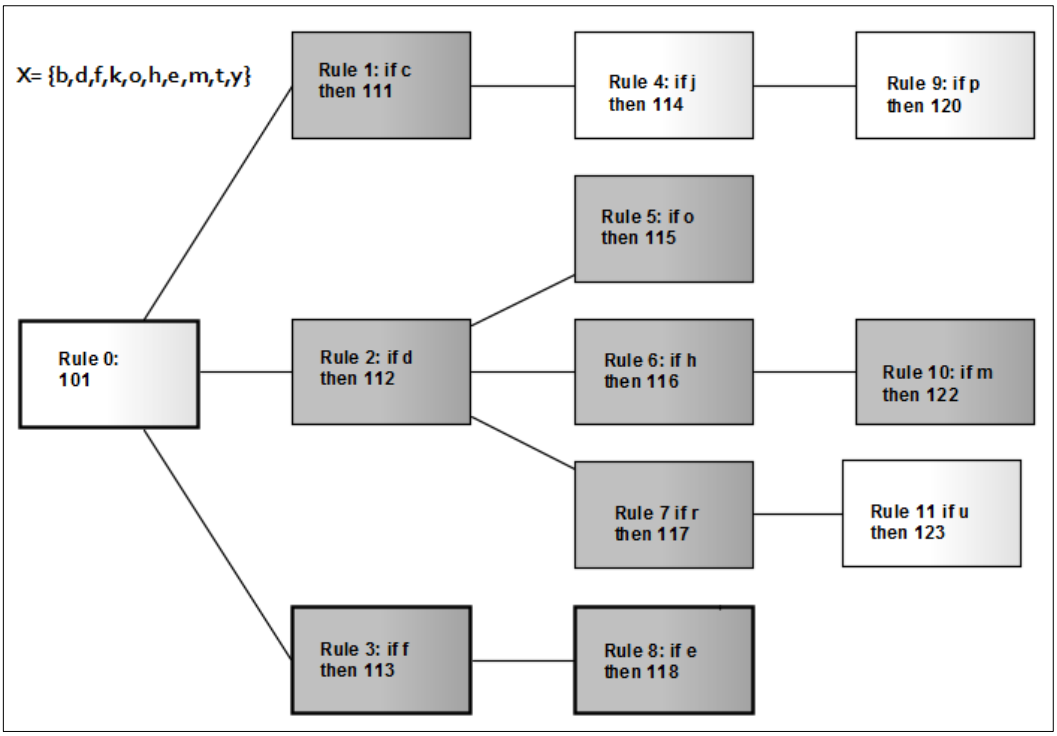


Figure 2.3: MCRDR Structure (Maruatona, 2013)

2.5.2 Prudence Analysis (PA)

Brittleness is a known issue in KBS. This is a situation where the ES does not realise when its knowledge is inadequate for a particular case. To address these issues, Prudence Analysis (PA) is introduced by (Compton, Preston, Edwards, & Kang, 1996). (R. Dazeley & Kang, 2008; R. P. Dazeley, 2006) also indicates that PA is a practical and extremely creative approach for solving the problem of brittleness in KBS.

(R. Dazeley & Kang, 2008) present another approach at PA where RDR was used to divide data into discrete subspaces. In the homogenous regions, an OD algorithm was used to identify anomalies. Their approach is split into three main tasks; profile retrieval, profile matching and classification. Profiles are kept in a Ripple Down Model (RDM) KB and RDM inference process retrieves a given profile. RDM is a modified variant of RDR where the resulting part of the (IF ELSE) rule is a profile rather than a conclusion or classification as is normal in RDR.

(R. Dazeley, Warner, Johnson, & Vamplew, 2010; Kelarev et al., 2012; Kim, Compton, & Kang, 2012; Sarawat, Yang, Byeong, & Qing, 2015) recognise that RDR has been implemented effectively in many functional applications of the KBS. Kim et al. proposed a Hybrid-RDR solution by integrating decision tree, J48 and censored production rules-based RDR. They proposed a schema mapping concept but addressed the problem of un-classification and incorrect classification in general. One of the concepts in PA is proposed by (R. Dazeley & Kang, 2008); it is an application of Rated-MCRDR (RM). It combines MCRDR with ANN.

2.5.3 Integrated Prudence Analysis (IPA)

A prior research on FD in online banking with IPA is done by (Maruatona, 2013), which says that commercially applied online payment FD systems have a common approach, which is the use of an RBS combined with an ANN. He further indicates that a prudent RDR system is a viable alternative in online banking FD. The author developed the IPA system from a selective combination of the best features of an attribute-based prudence method (MC-RDM) and a structural-based prudence method (RM). IPA proposes three combination techniques of RM and MC-RDM: IPAOR, IPAAND and IPAANN. The IPAOR/IPAAND is a result of combining RM's ANN output with the MC-RDM's aggregated outlier index through an AND or OR connection. When combined with the MCRDR indexes in IPAANN, the aggregated outlier index of MC-RDM outlier detectors is fed into the ANN. In these IPA techniques, Outlier Estimation with Backward Adaptability (OEBA) and Outlier Estimation for categorical attributes (OECA) are outlier detectors. A situated profile (SP) (Vastenburg, 2004) is used between MCRDR engine and the outlier detectors. OEBA is the method for numeric profiles, while OECA is for categorical attributes. The author further elaborates that the OEBA algorithm depends on the probability to model a continuous attribute in a dynamic environment, whereas OECA algorithm is used to detect outliers in categorical profiles.

2.5.4 RIDOR – A Ripple Down Rules Classifier

RDR is one of the well-known rule-based classification technique and is developed as an alternative to the traditional KBS (Compton & Jansen, 1988; Kang et al., 1995). (Richards, 2009) acknowledges that RDR is ideal due to its less maintenance and incremental learning capabilities. RDR significantly reduces the time and effort required to make the alteration

and ensures the consistency of the rulesets. (Kang et al., 1995; Richards, 2003) have highlighted that RDR systems have been used in many applications and classification domains. (Compton, 2011) acknowledges that RIDOR is most widely used RDR machine learner. Figure 2.4 shows a ruleset produced from RIDOR.

```
Class = Anon (1755.0/1582.0)

Except (Brows = Alt) => Class= Non (541.0/0.0) [256.0/0.0]

Except (Net_Cnt > 11) and (Log_Cnt <= 74) => Class=Fraud (36.0/0.0) [21.0/2.0]

Except (Log_Cnt <= 11.5) and (Acc_Type = PA) and (Src_Acc = Home_Loan) and (Src_Amt > 1900)
=>Class=Fraud (9.0/0.0) [3.0/0.0]

Except (Log_Cnt <= 10.5) and (Log_Cnt > 5.5) and (Src_Amt <= 1490.5) and (Acc_Type = PA) and
(Net_Cnt <= 3.5) and (Log_Cnt <= 7.5) and (Src_Amt <= 1139.95) =>Class=Fraud (10.0/0.0) [7.0/5.0]

Except (Log_Cnt <= 48) and (Log_Cnt > 19.5) and (Net_Cnt <= 7.5) and (Net_Cnt > 6.5) and (Src_Amt >
800) =>Class=Fraud (11.0/0.0) [6.0/1.0]

Except (Net_Cnt <= 5.5) and (Log_Cnt <= 51) and (Log_Cnt > 26.5) and (Src_Amt <= 1150) and (Log_Cnt
<= 30) =>Class=Fraud (16.0/0.0) [5.0/1.0]

Except (Log_Cnt <= 10.5) and (Src_Amt > 715) and (LogTime = PM) and (Log_Cnt > 5.5) and (Src_Amt
<= 1497.5) and (Src_Amt > 1300) => Class=Fraud (6.0/0.0) [2.0/0.0]

Except (Log_Cnt <= 8.5) and (Src_Amt > 1957.795) and (Log_Cnt > 6.5) and (Net_Cnt <= 3.5)
=>Class=Fraud (9.0/1.0) [2.0/0.0]

Except (Log_Cnt <= 8.5) and (Net_Cnt > 4.5) and (Log_Cnt <= 5.5) => Class=Fraud (4.0/0.0) [2.0/0.0]

Except (Brows = Moz_4) =>Class=Non (336.0/0.0) [183.0/0.0]

Except (Log_Cnt <= 2.5) and (Src_Amt > 1962.5) and (Net_Cnt > 1.5) and (Trn_Amt > 6480.32) =>
Class=Fraud (11.0/0.0) [9.0/3.0]

Except (Net_Cnt <= 2.5) and (Acc_Type = PA) and (LogTime = PM) and (Net_Cnt > 1.5) and (Country =
AU) =>Class=Fraud (17.0/1.0) [16.0/2.0]
```

Figure 2.4: RIDOR Ruleset for Bank Dataset

A sample RIDOR generated ruleset generated for Bank dataset is given above. It generates a default rule first (In this case is for Anon class label) and then the exceptions for the

default rule. It then produces the finest exceptions for each exception and expands cases like a tree. The exceptions are for all the rules for class prediction other than default class (for Anon class in this case). Incremental reduced-error-pruning is used to generate exceptions (Fürnkranz & Widmer, 1994).

Weka (Waikato, 1993) has RIDOR classifier as one of RDR implementation. There is also a Weka-based MapReduce application as Hadoop (ASF, 2015) wrappers, which can be used for classifying large datasets. However, (Meng et al., 2016; Shanahan & Dai, 2015) recognize Spark advantages over standard MapReduce. Spark retains the linear scalability and fault tolerance of MapReduce and is about 100 times efficient than MapReduce. Mahout is another Big data ML platform. But (Meng et al., 2016) highlight that Mahout is also MapReduce centred and that Spark's efficiency and scalability have been found to be higher than Mahout.

2.6 Conclusion

Fraud is growing every year, resulting in billions of dollars being lost by the businesses. FD for online banking is a significant area of research, but this study highlights a variety of issues. Both banks and IT services businesses do not release safety data information. Research on FD is also hard because exchanges of thoughts on the identification of fraud are very restricted. Banks do not supply information, or the supplied information is in small amounts, or may not provide particular characteristics to evaluate new studies. Additional issues include the heterogeneous nature of transaction records, scalable information and design efficiency, in particular the accuracy of the classification.

Considering the data constraints, synthetic data generation is a feasible option. An advanced method to generate simulated online banking transactions from limited reference data is suggested. For mixed data constraint, it is suggested that categorical characteristics should be converted into numerical attributes and the compacted sparsity. A proposed method for improving model efficiency involves FE and compact unified dataset expressions using profile models strategy. To strengthen classification accuracy, a single Unified Expression Ripple Down Rules FD methodology for Big data is suggested.

We propose an improved FD system in online banking with high volume, distributed as well as mixed dataset (dataset having numeric as well as categorical attributes). Figure 2.5 explains the complete process of FD for online banking.

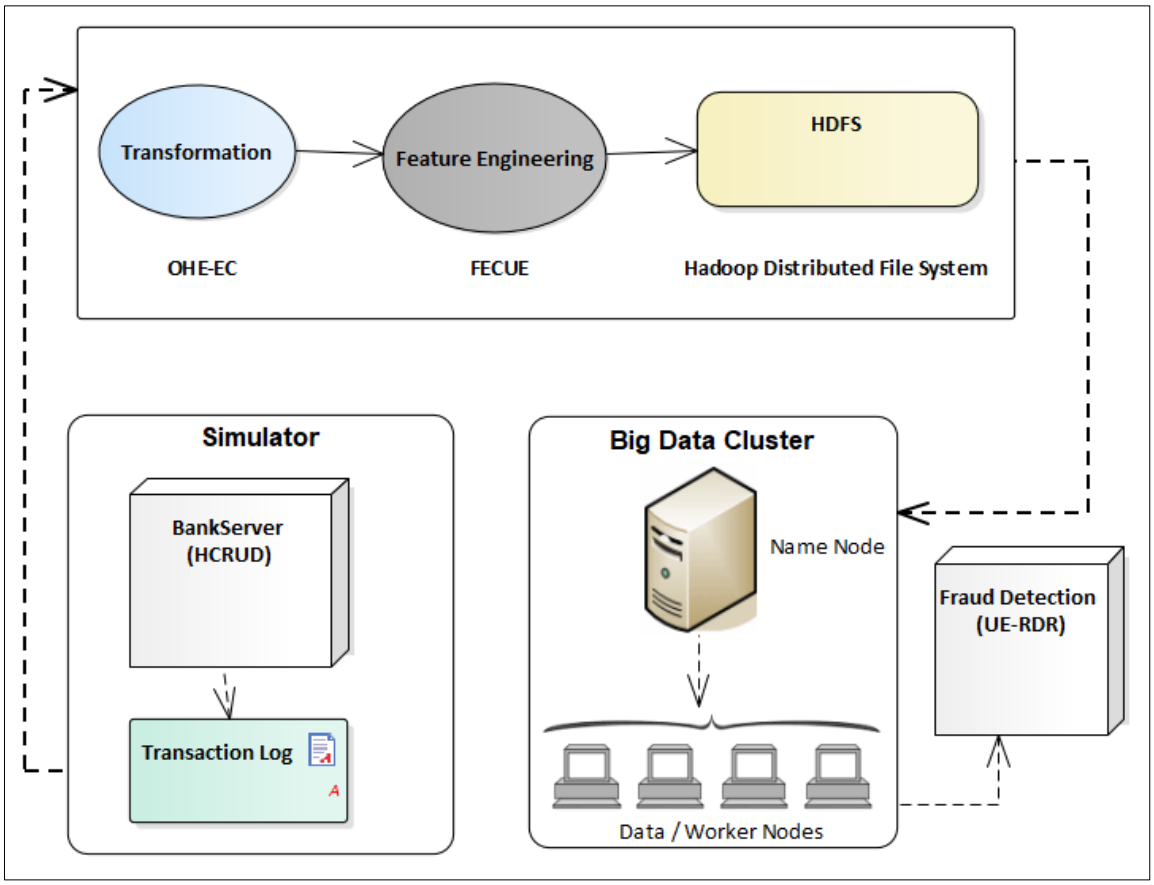
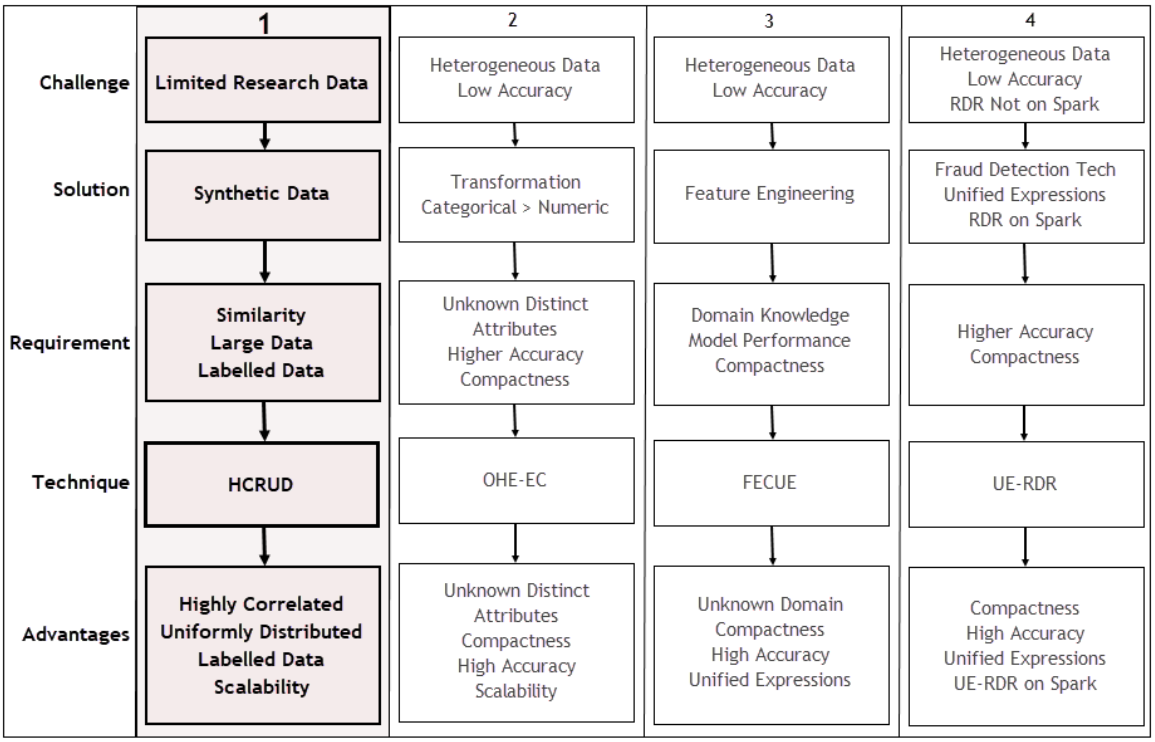


Figure 2.5: Fraud Detection Process for Online Banking

This diagram explains a complete FD process and its different components. First part is the simulator for synthetic data generation. Second part is the categorical to numeric data transformation. Then comes FE component to improve model performance. Final part is the FD technique. Unified Expression Ripple Down Rules based Fraud Detection Technique (UE-RDR) is for Scalable and Distributed Data for high classification accuracy with the use of Apache Spark. A multi-node Hadoop cluster with Spark was needed to be setup to develop and evaluate this system. The different roles in the cluster include spark gateway, history server, data nodes, node manager, name node and resource manager.

Chapter 3

Generating Synthetic Datasets for Experimental Validation of Fraud Detection



Chapter Overview

A summary of the techniques for detecting fraud was presented in the literature review in the previous chapter. It also provides an overview of different challenges in research into FD and underlines the research gaps. One of the most significant problems for academic research in FD is lack of access to large, real (or at least realistic) datasets for the development and evaluation of novel FD methods. In this chapter, this issue is addressed by HCRUD; it is an advanced technique to produce highly correlated synthetic data based on uniformly distributed RDR ruleset. The distribution of class labels, individual and the combination of correlated attributes are maintained in the generated data as per reference data.

The role of the work within this chapter within the overall research program is illustrated by the first highlighted block from the figure. It demonstrates the research problem and the approach addressed in this section. The figure challenge, solution, requirement and the developed technique are related to each other shows how challenge, solution, requirement and the developed technique are related to each other. This chapter describes the different methods of validation, as well as the features of the developed technique. This chapter discusses the fact that banks often supply research data, but the datasets are usually either low in size or may not contain specific characteristics needed to validate new techniques and algorithms. It presents a feasible approach to produce synthetic data in order to address this limitation. It also explains existing work carried out on synthetic data and the gaps in creating scalable synthetic data that maintains distribution class labels and on single as well as on combined attributes. It explains highly correlated synthetic data technique (HCRUD), based on a uniformly distributed ruleset. This chapter also provides an overview of the comparison between the data generated and the reference data and the empirical evaluation using RMSE and classification accuracy.

The work in this chapter was published as Ul Haq, I., Gondal, I., Vamplew, P. (2016). Generating Synthetic Datasets for Experimental Validation of Fraud Detection, 14th Australasian Data Mining Conference, Canberra, Australia. Conferences in Research and Practice in Information Technology, Vol. 170.

3.1 Introduction

Online banking frauds are resulting in billions of dollars losses to the banks around the world. Phishing related Internet banking frauds cost banks more than US$3 billion globally (McCombie, 2008). MCSI survey has highlighted that the annual worldwide impact of phishing and various forms of identity theft could be as high as US$5 billion and the cost of repairing damage to peoples' online reputation is much higher at around US$6 billion, or an estimated average of US$632 per loss (Marican & Lim, 2014). FD for online banking is a very important research area, but there are a number of challenges facing research on this topic. In particular knowledge on banks' FD mechanism is very limited and banks do not publish statistics of the FD systems (Maruatona, 2013). Most of the security is provided by third-party IT-companies who also protect their intellectual property from their competitors. So both banks and IT security companies do not publish information on their security systems. (Bolton & Hand, 2002) also highlight that development of new FD methods is difficult because the exchange of ideas in FD is very limited, but authors also support the notion that FD techniques should not be described with details publically; otherwise criminals may also access that information.

To conduct innovative research in fraud analysis, a large amount of data is required. Banks do provide data in some cases, but the data is normally either in small volume or may not provide specific features which are needed to verify new research techniques and algorithms. With the consideration of these limitations, a viable alternative is to generate synthetic data. This chapter presents an innovative technique for generating simulated online banking transaction data and evaluates how well this simulated data matches the original, small set of reference data. Further, the chapter presents FD study on the synthetic data.

Synthetic data can be used in several areas and the benefits of synthetic data are well presented by (Bergmann, 2015):

- It allows controlling the data distributions used for testing. So, the behaviour of the algorithms under different conditions can be studied.

- It can help in performance comparison among the different algorithms regarding the scalability of the algorithms.
- It creates instances having the finest level of granularity in each attribute, whereas in publicly available datasets anonymization procedures are applied due to privacy constraints.

3.2 Related Work

The idea of synthetic data generation is not new, as in 1993, Donald B. Rubin has generated data to synthesize the Decennial Census long form responses for the short form households (Rubin, 1993). However, it has not been applied to the area of online banking fraud.

Various attempts have been made to generate synthetic datasets. One technique uses uni-modal cluster interpolation e.g. SVDI (Coyle et al., 2013). This technique presents a method that uses data clusters at certain operating conditions where data is collected to estimate the data clusters at other operating conditions, thus enabling classification. This approach is applied to the empirical data involving vibration-based terrain classification for an autonomous robot using a feature vector having 300 dimensions, to show that these estimated data clusters are more effective for classification purposes than known data clusters that correspond to different operating conditions. SVDI's main shortcoming is that the estimates of data clusters and known data clusters have the same number of samples.

Different frameworks to synthesise the data (Bergmann, 2015; Keen, 2015; Maj, 2003; Wisser, 2015) have been studied, but all of these frameworks neither classify the data nor are based on any existing datasets. One attempt is to generate synthetic census-based micro-data with the customization and using extensibility of an open-source Java-based system (Ayala-Rivera et al., 2013). In the data generation process, they used probability weights by capturing frequency distributions of multiple attributes. Due to attribute interdependency, they also applied attributes constraints, but they have not applied the weightage on the combination of attributes. It might be possible that the distribution of individual attributes is the same in the generated data as in the reference, but this distribution cannot be guaranteed for the combination of the attributes. The generated data

cannot be used in the domain of classification problems, as this is not the classified data. Another attempt is made to generate constraint-based automatic test data. The technique is based on mutation analysis and creates test data that approximate relative adequacy (Demilli & Offutt, 1991). This technique is used to generate test data for unit and module testing. This work does not mention whether this technique is also applicable to produce data for classification.

(Chawla et al., 2002) present synthetic minority over-sampling technique, which is based on the construction of classifiers from imbalanced datasets. Under-sampling of the majority (normal) class has been proposed as a good means of increasing the sensitivity of a classifier to the minority class. Their method of over-sampling the minority class involves creating synthetic minority class examples. This approach is ideal for imbalanced data where the requirement is to reduce the majority class and increase the minority class. This technique is not ideal for increasing overall data size.

In another paper (Yoo & Harman, 2012) have proposed a technique to generate additional test data from existing reference data. Their paper highlights that mostly existing automated test data generation techniques tend to start from scratch, implicitly assuming that no pre-existing test data is available. However, this assumption may not always hold, and where it does not, there may be a missed opportunity; perhaps the pre-existing test cases could be used to assist the automated generation of additional test cases. The authors have used a search-based test data regeneration technique; that can generate additional test data from existing test data using a meta-heuristic search algorithm (Yoo & Harman, 2012). But the generated data, cannot be utilized in the domain of classification problems, as it does not have classification labels.

Another synthetic data generation and correlation technique is by (Christen & Vatsalan, 2013), which generates data based on real data having the capability to produce data for unicode character sets as well. This technique also caters attribute distribution and dependency. Besides these features, this technique is lacking labelled data and attribute distribution of multiple attributes. One novel technique is to generate synthetic data for

electronic medical records proposed by (Buczak, Babin, & Moniz, 2010). However, this technique can generate data mainly for the medical domain having the laboratory, radiology orders, results, clinical activity and prescription orders data elements.

In this chapter, an innovative technique has been presented, which generates highly correlated rule-based uniformly distributed synthetic data for fraud analysis. Empirical results are presented by comparing the generated data and original reference data. We have compared the distribution of individual attributes and combinations of correlated attributes. Classification accuracy results for FD are also observed with four well-known classification techniques. The empirical results show that the synthetic data preserves similar characteristics as the original reference data and have similar FD accuracy.

KBS can represent knowledge with tools and rules rather than via code. Mostly currently used FD systems use KB in their architecture with rule-based as a commonly used approach. RDR is suggested by (Compton & Jansen, 1988) as a solution to maintenance and KA issues in KBS. RDR is an approach to KA. RDR has notable advantages over conventional rule-bases; including, better, faster and less costly rule addition and maintenance approaches. Another benefit is the addition of PA of RDR systems which allows the system to detect when a current case is beyond the system's expertise by issuing a warning for the case to be investigated by the human. PA is introduced by (Compton et al., 1996).

The synthetic data generation approach can be used to generate data for any classification domain, but in this chapter, test data has been generated to simulate bank transactions to study fraud analysis in banking domain.

In the remainder of the chapter, section 3.3 presents our methodology in detail, while section 3.4 presents empirical results to show the working of the proposed technique. Finally, chapter is concluded in section 3.5.

3.3 Synthetic Data Generation Using Highly Correlated Rule Based Uniformly Distribution (HCRUD)

Synthetic data is generated with the following desired characteristics:

- In some attributes, the generated values should have constraints due to the attribute interdependency on those attributes.
- The continuous attributes values should be within predefined ranges set in the constraints.
- Single attributes should have similar attribute distributions.
- Paired attributes should have similar attribute distribution as the reference data.
- Data should have classification labels.

A high-level flowchart is given in Figure 3.1. The process is explained in Algorithm 3.1.

ALGORITHM 3.1: Transformation and Compactness

Input: Reference dataset
Output: Synthetic dataset
Begin

Step 1	Load Reference data in a two-dimensional matrix using Eq. 3.1
Step 2	Check attribute interdependency. Calculate attributes and class distributions from Reference Data using Eq. 3.2.
Step 3	Generate the Ruleset
Step 4	Start New Instance
Step 5	Generate attributes values from 1 to n with discrete probability distributions using Eq. 3.4.
Step 6	Validate generated attributes values with the ruleset expressions. (if all expressions are not validated then ignore the instance) Goto Step 4
Step 7	If generated attributes are validated in Step 6 then assign the classification label to these attributes (if not classified ignore the instance) Goto Step 4
Step 8	Validate class distribution (if not within range ignore the instance) Goto Step 4
Step 9	Finalize the Instance.
Step 10	Repeat from Step 4 to 9 till required instance count matches.
Step 11	Store Generated Data.
Step 12	**End**

A true representation of a generated synthetic data can be ensured by generating RDR from reference data and then generating data samples ensuring the distribution of both individual attributes and combinations of attributes remain the same as in the sample reference dataset.

A uniform distribution is applied to the attributes to keep data similarity. An innovative HCRUD technique is proposed in this chapter to generate synthetic data with desired characteristics.

Reference data is a two-dimensional matrix as given in Eq. 3.1.

$$D_R = [\ d_{ij}\ ] \tag{3.1}$$

where D_R is reference data and *i* are the attributes from 1 to *n* and *j* are rows from 1 to *m*.

Due to attribute interdependency in some attributes, constraints are applied to those attributes. The probability distribution of attributes is calculated with the ratio of the instances having a particular attribute value over the total instances in the reference dataset.

$$P_i = | D^i_R | / | D_R | \tag{3.2}$$

where Pi is the proportional value of the attribute i and | DR | is the cardinality of DR, i.e. reference data and D^i_R are the instances having attribute *i*.

In the first step, reference data is loaded from the source in the form of a matrix. Attribute interdependency, attributes and class distributions are calculated in the 2nd step. In the 3rd step, rules are generated from the reference data. Instance creation is initiated in the 4th step. In step five attributes values are generated by applying attribute interdependency and the discrete probability distribution on single as well as the combination of attributes, which is calculated in step 2. In the sixth step, an instance is formed with the generated attribute values which are then validated based on the established rules, ensuring single and multiple attributes distributions resemble with reference data. The instance is ignored if the instance is not validated from all the expressions from the ruleset. In step seven, after validating the instance, a classification label is generated for instance. In step eight, it is ensured that class distribution in the generated datasets is also maintained by ensuring the class distribution is within the threshold values. The instance is also ignored if a particular class distribution exceeds the threshold, calculated in step 2. The instance is finalised in step 9, and the steps 1 to 9 are repeated until the desired instance count is reached. Figure 3.1 is showing a high-level flowchart of the data generation process.

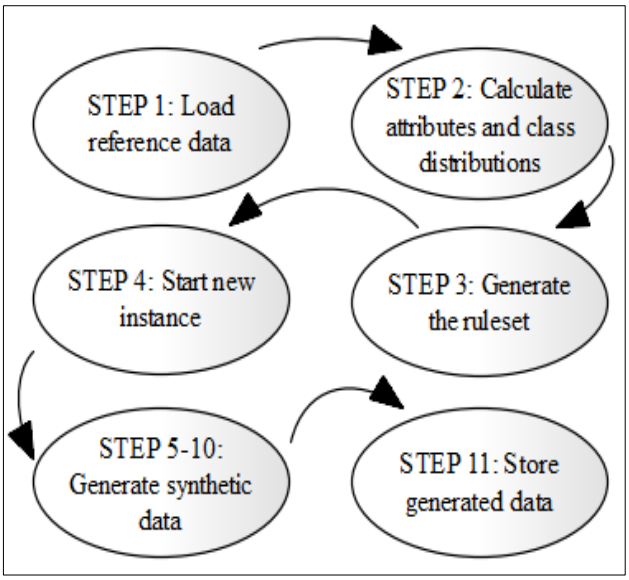


Figure 3.1: Synthetic Data Generation

The process of generating synthetic data is explained in detail in Figure 3.2.

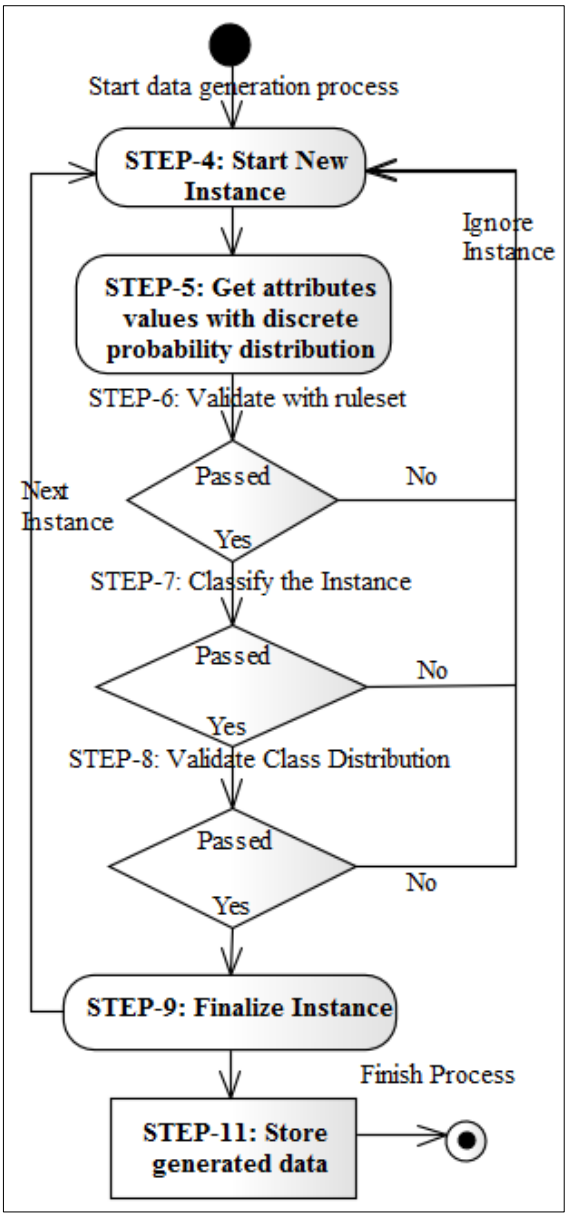


Figure 3.2: Detailed Process to Generate Data

3.3.1 Applying HCRUD to Generate a Synthetic Fraud Dataset

To evaluate FD algorithms in the banking data logs, a synthetic data emulating bank transaction has been generated, which is a mix of numerical and alphabetical attributes. An obfuscated dataset of 1775 internet banking transactions from a commercial bank was used to generate synthetic data. Although the dataset is small, the HCRUD technique presented in this paper demonstrates that a synthetic dataset can be generated of any desired size from small reference data. Format and structure of a typical online bank transaction dataset is

given in (Maruatona, 2013). The attributes of sample dataset are shown in Table 3.1. Different banks and FD systems adopt different nomenclatures for transactions.

Table 3.1: A Sample Bank Transaction Attributes

Name	Description	Type
Transaction ID	Unique ID for transaction	Label
Transaction Type	Type of transaction	Discrete
Account From	Source account	Label
Account To	Destination account	Label
Account Type	Type of account in use	Discrete
Event time	Time of transaction	Time
Session ID	Unique session ID	Label
Browser String	String describing browser	Label
IP Address	IP address for machine	Label
Country	Host country for given IP	Label
Trans Amount	Transfer amount (if Transfer)	Continuous
Biller Code	Unique biller code	Label
Biller Name	BPay Biller business name	Label
Log in ID	User's log in ID	Label
Log in Time	Time of log in	Time
Log in Count	Logins count for the day	Continuous
Password change	Password changes count	Continuous

Discrete probability distribution has been applied on the combination of attributes, i.e. transaction type and class to ensure close resemblance with the sample data:

$$F(x) = P(a \leq x \leq b) = \sum_{k=a}^{b} f(k) \tag{3.3}$$

where x takes value k between a and b. For the combination of the attributes, x is representing the combined value of the paired attributes Transaction Type and Class. Table 3.2 shows the distribution detail for the combination of attributes.

Table 3.2: Distribution of the Attributes for the Combination of Attributes

Transaction	Class	Probability
BPAY	Anon	0.022
BPAY	Fraud	0.083
BPAY	Non	0.208
PA	Anon	0.076
PA	Fraud	0.226
PA	None	0.386

where PA is 'Pay Anyone' and BPAY is a transaction type through which utility bills and other service providers can be directly paid. The class attribute represents the classification of Anon as anonymous and None as not a fraud. Only one combination of paired attributes is shown as an example here. More paired attributes, even more than two attributes can also be taken, but the more attributes we add, the more would be the ignored instances as mentioned in step 6 in Algorithm 3.1; hence it will take more time to generate the synthetic dataset. Experimental evaluation has shown that there are about 0.1% to 0.12% ignore cases by taking one combination of paired attributes.

Similarly, a discrete probability distribution is applied on individual attributes, i.e. transaction type and class separately, as shown in Eq. 3.4.

$$\sum_{k=a}^{b} f(k)=1 \tag{3.4}$$

Table 3.3 and Table 3.4 show the distribution details for single attributes.

Table 3.3: Single Attribute Distribution for Transaction Type

Transaction Type	Probability
BPAY	0.313
PA	0.688

Table 3.4: Single Attribute Distribution for Account Type

Account Type	Probability
Business	0.227
Other	0.001
Personal	0.773

Sum of the probabilities for both individual attributes is 1.0 Transaction Type and Account Type are the most significant attributes, so distributions detail of these two attributes is discussed above as an example.

3.3.2 Classification Techniques Used for Data Validation

The system is trained with generated datasets and tested on bank dataset. Datasets of different sizes were generated, ranging from 5,000 to 1 million transactions; detail is given in Table 3.8. Classification accuracy of the generated dataset is observed and compared with four well-known classification techniques: Decision Tree, RDR, Naïve Bayes and Random Forest (Breiman, 2001; Compton & Jansen, 1988; Quinlan, 1992; Richards, 2009; Swain & Sarangi, 2013).

3.3.2.1 Instance-Based Learning (IBL)

(Aha et al., 1991) have presented an IBL framework which generates classification predictions using only specific instances by applying similarity functions. IB1 and IBk are instance-based learners (IBL) (Chilo et al., 2009) which are also used for testing the classification accuracy in this chapter. IB1 is the simplest IBL and nearest neighbour-based algorithm where similarity function is used. It classifies the instance according to the nearest neighbour identified by Euclidean distance approach (Aha et al., 1991; Chilo et al., 2009). IBk is similar to IB1, but the difference is that in IBk, the K-nearest neighbours are used instead of only one. Three different distance approaches are employed in IBk, including Euclidean, Chebyshev and Manhattan Distance (Chilo et al., 2009).

3.3.3 HCRUD Implementation for Data Generation

Weka is a well-known data mining tool having a collection of ML algorithms and RIDOR is an RDR implementation in Weka. In this chapter, RDR ruleset is generated by using RDR classification from RIDOR:

$$R_R = \text{funcC}(D_R) \tag{3.5}$$

where RR is set of RDR format ruleset obtained by RDR classification function funcC. When reference data D_R is classified with RIDOR in Weka, it not only classifies the data but also generates a ruleset in RDR format.

A sample format of RDR Learner ruleset is given in Figure 3.3 that is used in this technique to produce rules from reference data.

```
Except (Browser = Alt) => Class = Fraud (546.0/0.0) [252.0/0.0]

Except (Network_Count <= 6.5) and (Transfer_Amt > 277.75) => Class = Non (37.0/0.0) [14.0/0.0]

Except (Network_Count <= 11) and (Login_Count > 11.5) => Class = Non (41.0/1.0) [31.0/1.0]

Except (Source_Acc = Business) and (Network_Count > 8) => Class = Non (4.0/0.0) [1.0/0.0]

Except (Network_Count <= 2.5) and (Acc_Type = PA) and (LogTime = PM) => Class =Fraud (28.0/4.0) [7.0/1.0]
```

Figure 3.3: A Sample of an RDR Ruleset

JEXL name stands for Java Expression Language, an implementation of Unified Expression Language (UEL) (ASF, 2004), JEXL is used to get the advantage of extra operators which are used in the rules compactness and to facilitate the implementation of dynamic and scripting features in this technique. The ruleset is transformed from RDR format to JEXL format, attributes-distributions and weightage calculated from reference data are fed to the proposed technique to generate the synthetic data. Figure 3.1 shows the abstract representation of the technique, while Figure 3.2 shows the detailed working of the synthetic data generation process. For compactness and efficiency, the generated rules are transformed to (JEXL) format:

$$R_J = \text{func}T(R_R) \tag{3.6}$$

where R_J is JEXL format ruleset and R_R is set of RDR rules and func*T* is transformation function of RDR ruleset.

A typical sample of JEXL expressions is shown in Figure 3.4.

```
Network_Count > 10 & Network_Count <= 12

Transfer_Amt > 2990 & Browser = Moz_4 & Country = AU

Login_Count <= 3 & Country = UK

BPay_Amt > 4750 & Browser = Moz_5Win & Country = AU

Transfer_Amt > 1005.5 & Browser = Opera

Acc_Type = BPAY & Source_Acc = Credit & Browser = Moz_4

PwdChange > 1 & Browser = Moz_5Win
```

Figure 3.4: JEXL Expressions Sample

Single classification, JEXL based implementation of RDR is developed and used in this technique to generate class labels to each generated instance. HCRUD generates dataset in a variety of formats including comma separated values (CSV), LibSVM(Chang & Lin, 2011) and Attribute-Relation File Format (ARFF) (Durrant et al., 2018), which are widely used data formats in any data mining and ML tools. A CSV format data is shown in tabular form in Table 3.5 as an example.

Table 3.5: CSV Format Example Dataset

Transaction Type	Amount	Account type	Pwd Changes	Login Time	Browser String	Country	Class
PA	4,000	Other	1	AM	Alt	Other	Non
BPAY	1,200	Personal	0	AM	Alt	Other	Non
PA	3,000	Business	0	AM	Moz_4	AU	Fraud
PA	4,000	Personal	0	AM	Alt	Other	Fraud
BPAY	860	Personal	0	AM	Opera	AU	Non
PA	1,500	Personal	3	AM	Moz_4	AU	Fraud
PA	1,422	Personal	0	AM	Alt	Other	Non

3.4 Results

After generating the datasets, the next step was to compare it with original reference data as a benchmark using two different measures. One of the measures was to check the

attribute distributions in the reference and generated datasets. Distributions of the individual as well as the combination of correlated attributes were also verified, including class association. The second measure was to check the classification accuracy in terms of FD by loading the generated data as training data and reference data as test data. Classification accuracy is verified in Weka with four well-known classification techniques including C4.5/J48, RDR/RIDOR, Random Forest and Naïve Bayes. IBL classification algorithms (IB1 and IBk) were also used to further verify the classification accuracy outcomes.

3.4.1 Quality Metric for Attribute Distribution

RMSE is a good accuracy measure and is also a commonly used measure for differences between values. Root mean squared error (RMSE) is used here as a quality measurement indicator, by taking the square root of the mean of the square of all of the errors for data distributions for individual and the combination of attributes. It is represented in Eq. 3.7.

$$RMSE = \sqrt{\frac{1}{N} \sum (D_R - D_G)^2} \tag{3.7}$$

where D_R is reference data and D_G is generated data.

3.4.1.1 RMSE for Combination of Attributes

RMSE for the distribution of individual attributes as well as the combination of attributes were calculated, and the experimental evaluation has shown that there is a minor difference in the attribute distribution of reference data and generated data.

The difference in data distribution for the combination of attributes in reference and generated datasets is shown in Table 3.6.

Table 3.6: Error in Distribution for the Combination of Attributes

Transaction Type & Class	Error
BPAY/Anon	0.80
BPAY/Fraud	1.18
BPAY/Non	1.81
PA/Anon	0.80
PA/Fraud	1.22
PA/Non	1.85

3.4.1.2 RMSE for Individual Attributes

The difference in data distribution for individual attributes is shown in Table 3.7.

Table 3.7: Error in Distribution for Single Attributes

Attribute	Value	Error
Class	Anon	0.11
Class	Fraud	0.11
Class	Non	0.00
Transaction Type	BPAY	0.16
Transaction Type	PA	0.22
Account Type	Business	0.03
Account Type	Other	0.03
Account Type	Personal	0.12
Country	AU	0.05
Country	Other	0.11
Browser String	Alt	0.78
Browser String	Mozilla	0.78

3.4.2 Class and Attribute Distributions

Comparisons of the class distribution and distribution of individual as well as the combination of correlated attributes are excellent measures to check how close the generated data is to the original reference data. Fifty datasets were generated, and classification and distribution results were averaged and compared with the original reference data.

Figure 3.5 shows the comparison of distribution by class in the generated dataset and in reference dataset; which is very similar.

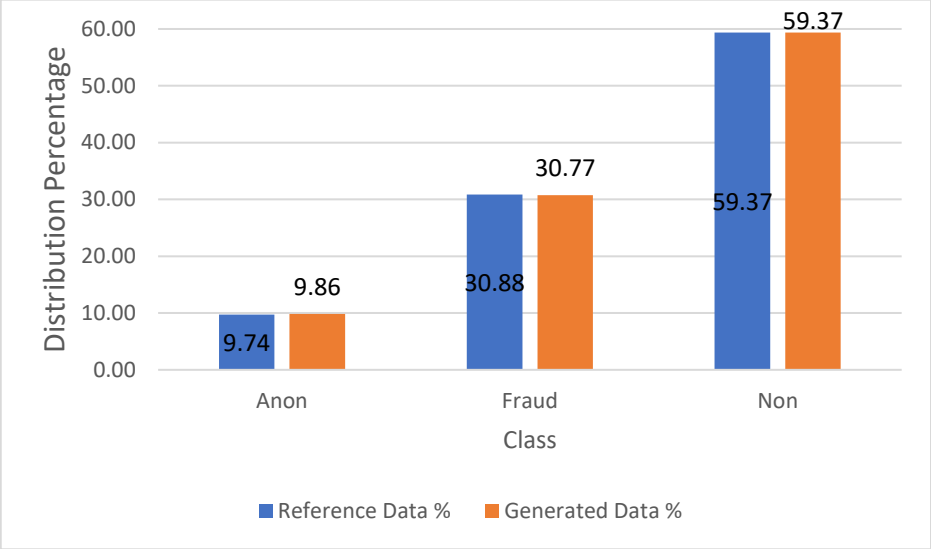


Figure 3.5: Distribution by Class

Figure 3.6 shows the comparison of the distribution of the combination of attributes (Transaction Type and Class) in the generated dataset and in the reference dataset. The results show that the percentages of values from both datasets are very close to each other.

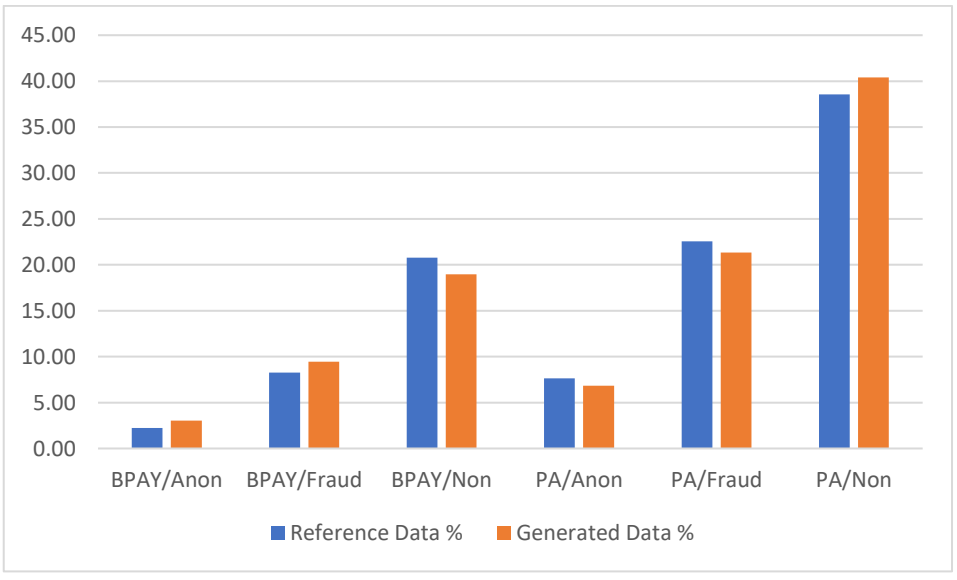


Figure 3.6: Distribution by Transaction Type and Class

Average time taken to generate instances is also calculated for the individual datasets. Results show that the average time taken to generate 1,000 instances is 2.67 seconds. Maintaining attribute and class distributions and assigning class labels to the instance are the few factors, due to which more time is being taken to generate the synthetic datasets. Figure 3.7 shows the time taken to generate each dataset. It also shows the trend line of time and data size.

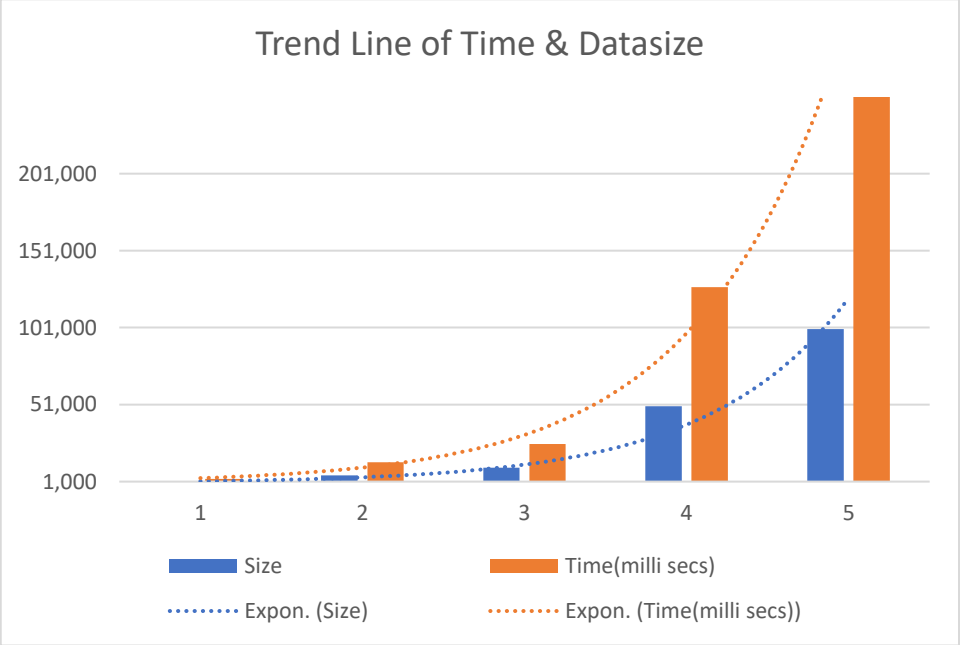


Figure 3.7: Time Taken to Generate Datasets

3.4.3 Comparing Classification Accuracy for Fraud Detection

Classification accuracy of the generated dataset is tested with four well-known classification techniques. Table 3.8, Table 3.9 and Table 3.10 contain the classification accuracy results; where generated data is used as training data while the reference data as test data using C4.5/J48, RDR/RIDOR, Naïve Bayes, and Random Forest classification techniques and IBL (nearest neighbour, similarity-based) algorithms as well. The mean classification accuracy for all generated datasets as well as the individual dataset is calculated and is very close to the individual accuracy percentage values.

Table 3.8: Fraud Detection Classification Accuracy Results

Dataset Size	RDR	C4.5	Naïve Bayes	Random Forest	Class. Mean
5k	72.19	75.27	65.70	71.34	71.13
10k	73.96	75.50	65.62	71.74	71.70
25k	76.58	76.24	65.19	75.24	73.31
50k	76.98	76.64	65.41	75.73	73.69
100k	76.98	76.81	65.36	77.09	74.01
500k	77.04	76.98	65.19	77.44	74.10
1mil	76.98	76.92	65.13	77.98	74.22
Dataset Mean	76.03	76.34	65.37	74.93	73.17

Classification accuracy results are showing that with the increase size of training data (generated data), there is an increase in the accuracy percentage in RDR, C4.5, Random Forest and Classification mean column as well.

Another testing is also performed using cross-validation with fold=1755 for both reference and generated data. Fold value of 1755 was taken, due to the reference data size of 1755 instances. Table 3.9 shows the classification result with four classification techniques with both reference data and generated data.

Table 3.9: Classification Accuracy Results with Cross-Validation

Classification	Reference Data	Generated Data	Difference
RDR	77.83	94.02	16.18
C4.5	87.41	96.70	9.29
Naïve Bayes	70.09	89.23	19.15
Random Forest	89.40	94.81	5.41

The results are showing that classification accuracy is higher when the system is trained on generated data.

To further verify classification accuracy with IBL (nearest neighbour, similarity-based) algorithms, we have performed the evaluation with IB1 and IBk algorithms. Classification accuracy results with IBL are presented in Table 3.10.

Table 3.10: Classification Accuracy Results with Instance-Based Learning Algorithms

	IBk	**IBk**	**IBk**	**IB1**
Dataset Size	Euclidean Distance	Chebyshev Distance	Manhattan Distance	
5k	65.64	64.50	66.84	66.95
10k	68.03	67.01	67.12	68.09
25k	71.19	69.18	72.42	72.29
50k	71.69	69.95	73.08	72.89
100k	72.59	70.71	73.73	73.11
500k	73.33	71.28	73.05	73.22
1mil	74.30	73.11	75.44	75.10

Classification accuracy results shown in Table 3.8 - Table 3.10 depict that with the increase of training data (generated data), there is an upward trend of the classification accuracy percentage.

3.5 Conclusion

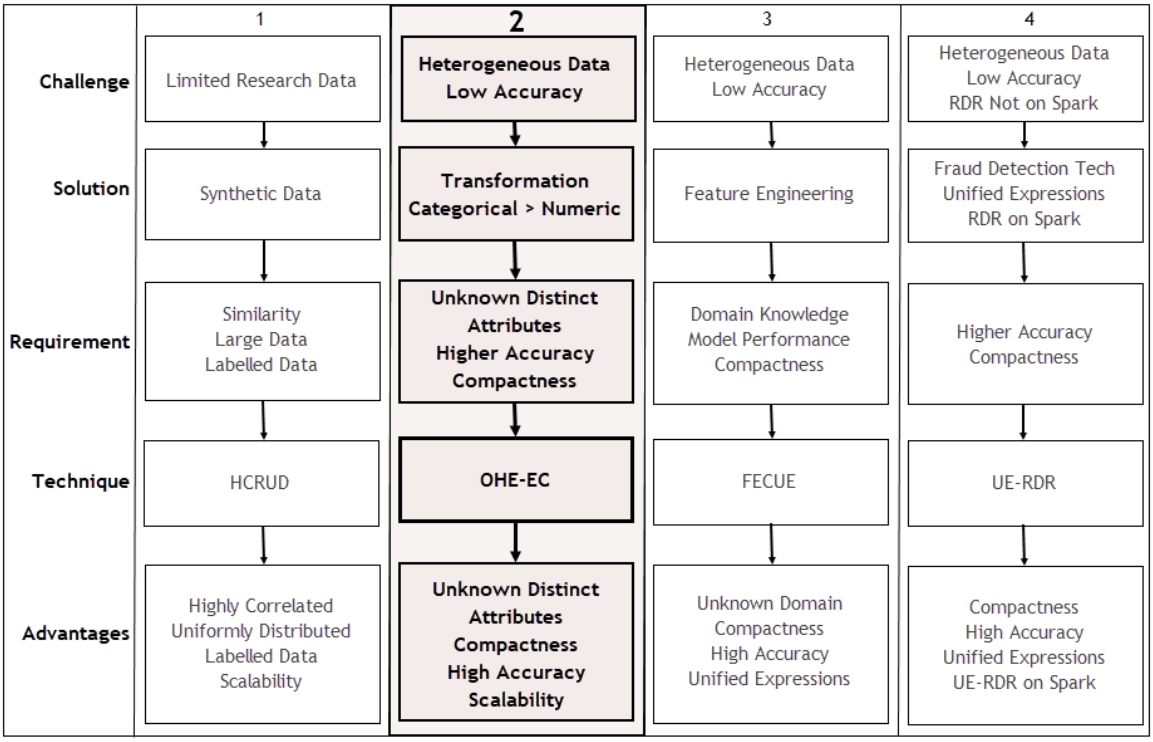
To overcome a challenge of limited availability of datasets for fraud analysis studies for financial institutions, an innovative technique: highly correlated rule-based uniformly distributed synthetic data has been presented to generate synthetic data. In this chapter, we have presented the comparison of the distributions of the original and the synthetic data and the comparison of FD classification accuracy with well-known classification techniques. A single classification, JEXL based Java implementation of RDR is developed and used to generate class labels to each generated instance. In classification accuracy testing, we used generated data as training and original data as test data. Empirical results show that synthetic dataset preserves a high level of accuracy and hence, the correlation with original reference data. Finally, we used an RMSE as a quality metrics for root mean square error to determine the difference of data distribution for individual and the combination of

attributes in generated datasets as compared to original reference datasets. Studies have shown very similar distributions of the attributes of generated datasets.

Currently, we are generating the dataset with only 13 attributes of an obfuscated dataset. It needs to be more efficient; otherwise, for high-dimensional data, it will take more time. One of the recommended future works is to test this technique on high-dimensional data, while another work is to handle missing values from the reference data.

Chapter 4

Categorical Features Transformation with OHE-EC for Fraud Detection in Distributed Environment



Chapter Overview

The previous chapter (Chapter 3) has presented an overview of synthetic data generation having scalability and high correlation from existing data. Categorical features transformation with compact One-hot Encoder using the OHE-EC technique is used in this chapter to address the problems of heterogeneous data in FD study.

The role of the study in this chapter is demonstrated by the second highlighted block from the figure in the overall research program. It demonstrates heterogeneous nature of data as the research problem and its solution. This chapter highlights why mixed data conversion is needed, and it describes the features of the developed technique. It explains data transformation from mixed to numeric (OHE-EC) technique and explains the two models in this technique. This technique transforms categorical features to numeric features by compacting sparse-data even if all the distinct attributes values are not known in advance. This chapter also provides an overview of empirical evaluation with classification accuracy using Random Forest, Decision Tree, Naïve Bayes, SVM and OneVsRest classifiers on Big data platform with multiple datasets.

The work in this chapter was published in: (Ul Haq et al., 2018) Categorical Features Transformation with Compact One-hot Encoder for Fraud Detection in Distributed Environment, Data Mining: 16th Australasian Conference, AusDM 2018, Bathrurst, NSW, Australia, Vol. 996.

4.1 Introduction

OD techniques have been in use for many applications including ID and FD (Breunig, Kriegel, Ng, & Sander, 2000; Hodge & Austin, 2004; Jin et al., 2010; Maruatona, 2013; Y. Zhang, Meratnia, & Havinga, 2010). Most of the OD methods use homogeneous datasets having a similar type of the attributes like numerical or categorical attributes, but real-world datasets often have a combination of these attribute types (K. Zhang & Jin, 2010). For example, section 3.3.1 and Table 3.1 shows that a typical bank transaction dataset has attributes which are a combination of numeric and categorical attributes.

Numeric features give better performance in classification and regression algorithms. Similarly, clustering algorithms work effectively on the data where all attributes are either numeric or categorical data, as most of the algorithms perform poorly on mixed data types (Shih et al., 2010). (Z. Huang, 1997) describes in his finding that clustering methods like k-means are efficient for processing large datasets, but these methods are often limited to numeric data. In addition, ML software may only support certain types of data. For example, Apache Spark (Meng et al., 2016; Pentreath, 2015; Shanahan & Dai, 2015) is a highly scalable platform to run ML algorithms in a distributed environment, but it accepts only numeric data for classification, regression and clustering algorithms. Therefore, there may be a need to convert categorical variables to a numerical encoding.

Categorical variables are commonly encoded using OHE. (W. Chen, 2016) indicates that in many traditional data mining tasks, OHE is widely used for converting categorical features to numerical features. OHE transforms a single variable with n observations and d distinct values, to d binary variables with n observations each. Each observation indicates the presence 1 or absence 0 of d^{th} binary variable. However, data becomes sparse after this transformation.

Sparse datasets are common in the Big data, where the sparsity comes from factors, i.e. feature transformation (OHE), large feature space and missing data (Meng, 2014). For a given attribute, OHE will increase the number of attributes from one to n distinct values in that attribute, which will not only make the datasets high-dimensional but also increase datasets size. (W. Chen, 2016) believes that other than the accuracy, due to growing memory and storage consumption, compactness of ML models will become equally important in the future.

We have presented a technique to transform categorical attributes to numeric attributes and compact the data sparsity. The transformed data can be used for the experimental validation and development of FD technique, especially for scalable and distributed data. This technique is tested on an FD bank data and on an AD KDD-99 dataset, which is widely used as one of the few publicly available datasets for AD (Tavallaee, Bagheri, Lu, &

Ghorbani, 2009). A multi-node Hadoop cluster is used for experiments, and the performance comparison of the technique has been presented with different classification techniques.

4.1.1 Contribution

Considering model accuracy and importance of growing memory and storage needs, we have developed a technique to transform categorical attributes to numeric attributes and compact the sparse data as well. An innovative technique is developed and presented in this paper to transform categorical features to numeric features by compacting sparse-data even when all the distinct values are not known in advance. Two further models are also developed in One-hot Encoding Extended Compact technique and classification accuracy is evaluated with both models.

Our main contributions in this research are summarized as follows:

- Developing One-hot Encoded Extended (OHE-E) technique.
- Extending One-hot Encoded Extended with Compactness (OHE-EC).
- Develop two further models: First Come First Serve (FCFS) and High Distribution First (HDF) in One-hot Encoded Extended Compact (OHE-EC).
- Evaluating classification accuracy, the effect on data size and efficiency in terms of the training model and prediction with well-known classification techniques.
- Empirical evaluation with a synthetic dataset generated from real bank transaction data and the well-known KDD-99 dataset.

4.2 Related Work

Several efforts have been made in the past to transform categorical attribute to numeric attributes. First attempt and one of the popular way to convert a categorical feature to a numerical is OHE, but this transformation results in high-dimensional sparse-data. (Jian et al., 2017) have transformed categorical data with CDE technique by extending coupling learning methodology by obtaining hierarchical value-to-value cluster couplings. CDE is slower than other embedding methods, thus is not ideal for large datasets. It is only applied

to unsupervised clustering domain. Another categorical data-representation technique is proposed by (Qian et al., 2016) with an objective of solving the problem of the categorical data not having a clear space structure. The authors have not addressed the problem of clustering for a mixed dataset. A comparative evaluation of similarity measures for categorical data is done by (Boriah et al., 2008). But the evaluation is performed in a specific context of OD, and relative performance of similarity measures is not studied for classification and clustering. (Boriah et al., 2008) highlight that several books on cluster analysis (Anderberg, 1973; Hartigan, 1975; Jain & Dubes, 1988) that discuss the problem of determining the similarity between categorical attributes, recommend binary transformation of data for similarity measures.

To overcome these limitations and for better accuracy, we have presented a technique to transform categorical attributes into numeric attributes and compact the data sparsity. This data can be used for the experimental validation and development of FD technique, to check scalability in a distributed environment.

4.3 Methodology

The bank, synthetic and KDD-99 datasets contain some attributes where distinct values are not always known in advance, so OHE was not ideal for these data sets. (Qian et al., 2016) technique is mainly for clustering domain, to solve the problem of categorical data without a clear space structure. CDE technique is not ideal for large datasets and is only unsupervised clustering domain, so it was not suitable for large synthetic data sets.

We have further extended Highly correlated rule-based uniformly distributed synthetic data (HCRUD) (Ul Haq et al., 2016) to generate numeric synthetic data from mixed reference data. A multi-node Hadoop cluster is used for experiments in a distributed environment with a name node, resource manager and multiple workers and data nodes. The complete process of loading data, filtering categorical features, distribution, transformation, and compactness is explained in the Algorithm 4.1.

4.3.1 Algorithm 4.1

#Load source data and perform Feature selection with Singular Value Decomposition (SVD) using Eq. 4.1.

#Filter categorical features only. Distribute data rows on worker nodes in the distributed environment in a multi-node Hadoop cluster using Eq. 4.4. Block size and replication factor is configurable. We have used 64-MB block size and three replication factor. Distributing data on worker nodes gives efficiency with data locality.

ALGORITHM 4.1: Transformation and Compactness

Input: Instance from a mixed dataset.
Output: Instance with the compact numeric format.
Begin

Process rows on worker nodes in parallel and Process each Row.

a. Process each Feature

b. IF (Feature is Selected and Categorical)

i. For each Feature transform with OHE-E adding extra feature using Eq. 4.5.

#Missing value imputation (MVI) is applied with the majority value of a given attribute for selected attributes. The decision of taking extra attribute is configured in various contextual and model-based profiles. It is evaluated with different measures explained in section 4.3.3.

ii. Check sparsity of the vector created with the transformation Step i using Eq. 4.2 & Eq. 4.3.

iii. Compact the sparse-data values using Eq. 4.6.

```
FOR Feature 1 to n LOOP
   IF feature NON-ZERO AND NOT NULL
      CompactFeature = featureIndex:feature
   ELSE
      SKIP VALUE
   NEXTVALUE
ENDLOOP
```

c. IF (more features in the row) Goto Step a

#Compact complete Row using compact values from Step a - c

```
CompactRow = EMPTY
FOR CompactFeature 1 to n LOOP
   CompactRow = CompactRow + SPACE + CompactFeature
      NEXTVALUE
ENDLOOP
CompactRow = ClassLabel + SPACE + CompactRow
```

#Map and reduce tasks are used for processing and resource manager manages the processing jobs.

#IF (more Row) from any worker node Goto Step 4 ELSE

FINISH

End

Source data can be represented in a two-dimensional matrix: $D_S = [\ d_{ij}\ ]$ where D_S is reference data and having i attributes from 1 to n and j are rows from 1 to m. Feature reduction is done using SVD, which is a well-known method used for dimensionality reduction. SVD factorizes a matrix into three matrices: U, Σ, and V.

$$A=U\Sigma V^T \tag{4.1}$$

where U is an orthonormal matrix, Σ is a diagonal matrix with non-negative diagonals in descending order, V is an orthonormal matrix and V^T is the conjugate transpose of V. Sparsity of a vector or matrix can be represented as:

$$V^S = \sum_{1\ (k=0)}^{n} / \sum_{1}^{n} \tag{4.2}$$

where sparsity is the ratio of the sum of attributes of a vector V from 1 to n having value k=0 to the total attribute values. The sparsity can also be represented as Eq. 4.3, which is 1 minus, the sum of the number of attributes which are non-zero.

$$V^S = 1 - \sum_{1\ (m\neq 0)}^{n} \tag{4.3}$$

where m are the attribute values, which are non-zero.

4.3.2 Data Blocks

When a file is stored in Hadoop Distributed File System (HDFS) (ASF, 2015), the system breaks it down into individual blocks set and stores these blocks in multiple slave nodes (worker nodes) in the Hadoop cluster. Rows division in each data block can be calculated with Eq. 4.4.

$$Rows^{Block} = \Sigma Rows/WorkerNodes\ /\ DataBlockSize/RowDataSize \tag{4.4}$$

4.3.3 Transformation with OHE-E

One-hot Encoding Extended (OHE-E) is a technique developed in this chapter, which transforms categorical attributes to numeric attributes with an extra attribute. Missing value imputation (MVI) is applied with the majority value of a given attribute for selected

attributes. Transformation with One-hot Encoding Extended with an extra attribute is explained in Eq. 4.5.

$$E^{ohe\text{-}e} = funcTrans(A^{d}) \quad (4.5)$$

where $E^{ohe\text{-}e}$ is One-hot Encoding Extended (OHE-E) format and A^{d} is attribute with d predefined distinct values and funcTrans is transformation function of OHE-E. funcTrans(A^{n}) function transforms a selected and categorical attribute A with n observations and d distinct attribute values, to d +1 binary attributes with n observations each. Each observation is indicating the 1 as true or 0 as false of the dth+1 binary variable. The dth+1 variable will be true if an attribute value is not from the predefined attributes values. The extra attribute is only included if there is a possibility of new values from previously known values. The decision of taking extra attribute is configured in various contextual and model-based profiles. It is evaluated with different measures including; the ratio of total d distinct values of an attribute with n observations. The threshold applied in bank dataset is 0.005. Another measure is time-bound attribute values. For example, in a banking application, the types of transactions can be enumerated in advance, but other attributes such as the device or browser being used may continue to exhibit novel values over time as technology changes. In bank dataset example, let us assume a categorical attribute with n observations and d distinct values. If in a particular row there is a new attribute value then the conversion with OHE will be represented as below with all columns as false.

col1	col2	col3	col4	…….. coln
0	0	0	0	……...0

However, the conversion with OHE-E will be represented as below with all columns as false, but n+1 column as true.

col1	col2	col3	col4	…….. coln	coln+1
0	0	0	0	…….. 0	1

OHE-E conversion example is also shown in Table 4.1

4.3.4 Compactness with OHE-EC

OHE converts categorical attributes into a format that better fits for algorithms of classification and regression (L. Zhang, Xiong, Zhao, Botelho, & Heffernan, 2017). However, transformation with conventional OHE generates sparse data with many 0 values (L. Zhang et al., 2017; Zhou & Xiang, 2015), so compactness of data is suggested and applied in this technique. Compactness on sparse-data is applied by omitting all zero and empty attributes values in an instance and keeping the remaining attribute values along with the attribute index. Compactness is explained in Eq. 4.6.

$$C^{ohe\text{-}ec} = funcCompact \int_{i}^{n_Y} (X)\ m \neq 0 \tag{4.6}$$

where X is $E^{ohe\text{-}e}$ format data from Eq. 4.5 and $C^{ohe\text{-}ec}$ is the OHE Extended Compact format and funcCompact is a function to compact a row y with only selecting attributes from 1 to n on i^{th} index having m value which is non-zero. The empirical evaluation has shown that after compacting data with OHE-EC, size could be 3x smaller from OHE format.

4.3.5 Sample Datasets Formats

A sample of the mixed datasets is explained by (Ul Haq et al., 2016), Table 4.1 shows sample data, in OHE format for categorical attributes; Transaction Type (BPay and PA), Account Type (Credit, Personal), Browser (Alt, Moz4, Browser New) and Country (AU, NZ, Country. New), while Table 4.2 shows compact OHE format for same data in Table 4.1. The compacting process is explained in Eq. 4.6.

Table 4.1: One-hot Encoding Extended Dataset

Class	Bpay	PA	Amount	Credit	Personal	Login	Password	Alt	Moz 4	Moz 5	Brows. New	AU	NZ	Count. New
1	0	1	8210	0	1	5	1	0	0	1	0	1	0	0
0	0	1	5124	0	1	4	1	0	0	1	0	1	0	0
2	0	0	2035	0	1	8	2	0	0	0	0	0	0	1

Table 4.2: Compact Data Format

Class	Attributes
1	2:1 3:8210 5:1 6:5 7:1 10:1 12:1
0	1:1 3:5124 4:1 6:4 7:1 9:1 13:1
2	2:1 3:2035 5:1 6:8 7:2 8:1 14:1

First Come First Serve (FCFS) and High Distribution First (HDF) are the two models in this technique. Eq. 4.5 explains that OHE transforms a single variable with n observations and d distinct values, to d +1 binary variables with n observations each. Each observation indicates the presence 1 or absence 0 of the binary variable. Distribution is calculated for a binary variable having the presence in n observations. In FCFS, no sorting is done, but in HDF, the attributes are sorted based on the distribution (higher distribution first). FCFS is efficient in training and testing the model, but it has relatively lower classification accuracy. HDF has better classification accuracy but is a little slower in training and testing due to the extra overhead of sorting higher distribution attribute values. The empirical evaluation has shown that if lower distribution attributes are excluded then accuracy with HDF further increases as compared with FCFS.

OHE-EC technique not only reduces dataset size but gives better performance also in terms of classification accuracy and time (especially on Hadoop multi-node cluster) and data can also be used in the classification techniques which use numeric data only.

4.4 Results

4.4.1 Synthetic Bank Transaction Dataset

A synthetic dataset based on actual bank transaction data was generated using the HCRUD technique (Ul Haq et al., 2016). Comparison of classification accuracy with synthetic generated mixed data (generated by HCRUD), and numeric data (converted by OHE) is shown in Table 4.3 and Table 4.4 for different classification algorithms Random Forest, Decision Tree, Naïve Bayes, SVM and OneVsRest (Breiman, 2001; Cortes & Vapnik, 1995; Quinlan, 1992; Sánchez-Marono, Alonso-Betanzos, García-González, & Bolón-Canedo, 2010; Swain & Sarangi, 2013). Training and test data split ratios is 70% and 30% respectively and average results are taken.

Table 4.3: Accuracy with Mixed Datasets

Random Forest	Decision Tree	Naïve Bayes	SVM	OneVsRest	Instances in Dataset
96.02%	97.55%	63.59%	60.99%	62.79%	10,000
97.77%	98.85%	64.39%	61.01%	62.58%	100,000
97.90%	98.84%	64.07%	61.57%	62.96%	1,000,000

Table 4.4: Accuracy with Numeric Datasets with OHE

Random Forest	Decision Tree	Naïve Bayes	SVM	OneVsRest	Instances in Dataset
97.93%	97.76%	64.86%	93.60%	94.12%	10,000
98.82%	98.85%	64.05%	93.04%	93.21%	100,000
98.88%	98.82%	63.95%	93.24%	93.66%	1,000,000

Classification accuracy results shown in Table 4.3 and Table 4.4 depict that classification accuracy is better with numeric data (OHE) as compared with a mixed dataset. A T-TEST was performed to determine whether classification accuracy in Table 4.3 and Table 4.4 are likely to have come from the same two underlying populations that have the same mean or those values have any significant difference. T-TEST, results prove that the classification accuracy results have significant differences. Standard deviation for multiple runs of 70%

and 30% data splits was also calculated. Deviations for mixed data set is 0.6581, 0.5594, 1.9487, 0.4537 and 0.3714 respectively, while the deviations for OHE is 0.4012, 0.5355, 0.3771, 0.2829 and 0.3593 respectively.

FCFS and HDF are two further models developed in One-hot Encoding Extended Compact (OHE-EC) technique. Table 4.5 and Table 4.6 show a comparison of classification accuracy with these two models.

Table 4.5: OHE-EC (FCFS)

Random Forest	**Decision Tree**	**Naïve Bayes**	**Instances in Dataset**
97.97%	97.67%	64.77%	10,000
98.84%	98.62%	63.98%	100,000
99.02%	98.95%	63.83%	1,000,000

Table 4.6: OHE-EC (HDF)

Random Forest	**Decision Tree**	**Naïve Bayes**	**Instances in Dataset**
98.16%	97.79%	63.29%	10,000
98.92%	98.76%	64.23%	100,000
99.07%	99.07%	63.84%	1,000,000

The classification accuracy results in Table 4.5 and Table 4.6 suggest that classification accuracy with OHE-EC (HDF) is slightly better than OHE-EC (FCFS). To confirm this a T-TEST was performed on these results. T-TEST results for Random Forest, Decision Tree and Naïve Bayes are 0.6075, 0.5162 and 0.2113 respectively, indicating that the observed differences between OHE-EC (HDF) and OHE-EC (FCFS) with regards to classification accuracy are not statistically significant. Standard deviations for FCFS model is 0.4589, 0.5514 and 0.6927 and for HDF is 0.4031, 0.5471 and 0.4144.

4.4.1.1 Parameters Selection

The different classifier used in the empirical evaluation was using different parameters used by the particular classifiers. For most of the classifiers, the default values of the parameters were used, but for some parameters, the optimal values of the parameters were used which were giving better results. Linear SVM model was used for SVM. In Random Forests num Trees = 150, feature subset strategy = all, impurity = gini, max depth = 30, max bins = 150 and seed = 12345 were used. In NaiveBayes model type = multinomial and lambda = 1.0. and in OneVsRest fit Intercept=True and tolerance = 1E-6 were used. In Decision Tree impurity (gini), 10 as Max depth and 150 as Max bins were used. While in Spark MR2 as YARN and submit replication =1, buffer size = 64 KB and client deploy mode parameters setting were used.

Other than the classification accuracy, one measure was to compare the model's training and prediction time with OHE and OHE-EC. Figure 4.1 shows training and prediction improvement with OHE-EC in terms of the time.

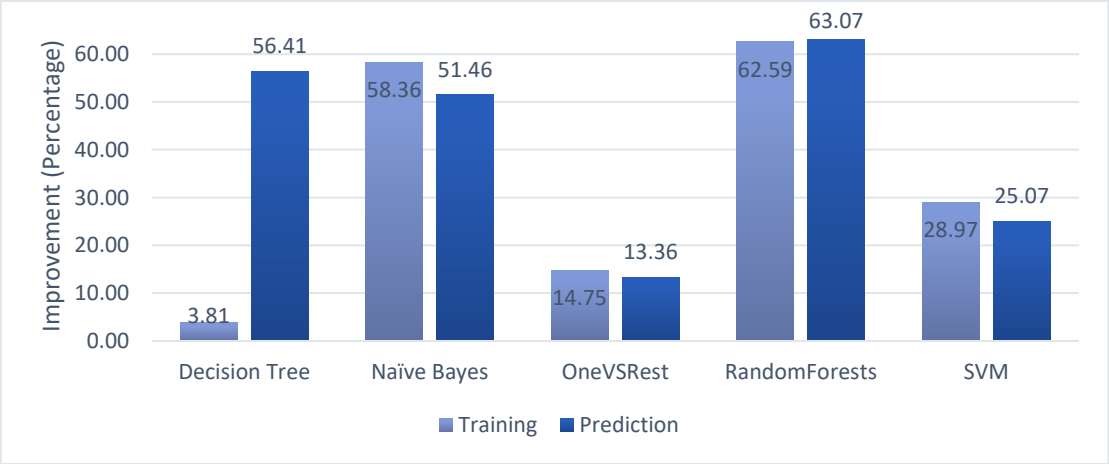


Figure 4.1: Average Train/Prediction Time Improvement with OHE-EC

X-axes in the above figure are the classifiers, while Y-axes are the average improvement time for different dataset size ranging from very small to large datasets. Results show that there is a significant improvement in training and prediction times of the models with OHE-EC. Another empirical evaluation was done with larger datasets only. Figure 4.2 shows that

improvement in prediction time is higher than the training time with larger datasets in almost all classifiers other than Random Forest.

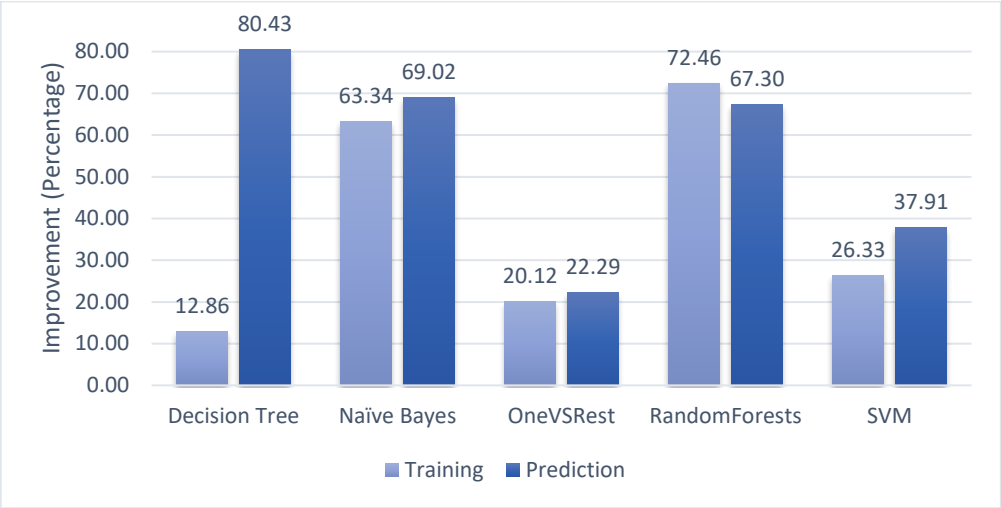


Figure 4.2: Large Data Train/Prediction Time Improvement with OHE-EC

4.4.2 KDD Cup Data

The proposed technique was also tested on a KDD-99 dataset, a widely used publicly available datasets for AD (Tavallaee et al., 2009). The current datasets contain more than 65 distinct attributes values in service attribute. There is a high possibility that there is a new service in the data. One-hot Encoding Extended can transform the row to OHE-E as it is using one extra attribute for new attribute values. Table 4.7 shows a comparison of classification accuracy with 10 million instances of KDD-99 dataset.

Table 4.7: Comparison of Performance of Various Classifiers on the KDD-99 Dataset

Random Forest	Decision Tree	Naïve Bayes	SVM	Format	Model
99.973%	99.920%	93.043%	99.991%	Mixed	
99.986%	99.997%	93.711%	99.990%	OHE	
99.99%	99.993%	93.265%	99.997%	OHE-EC	FCFS
99.993%	99.993%	93.463%	99.999%	OHE-EC	HDF

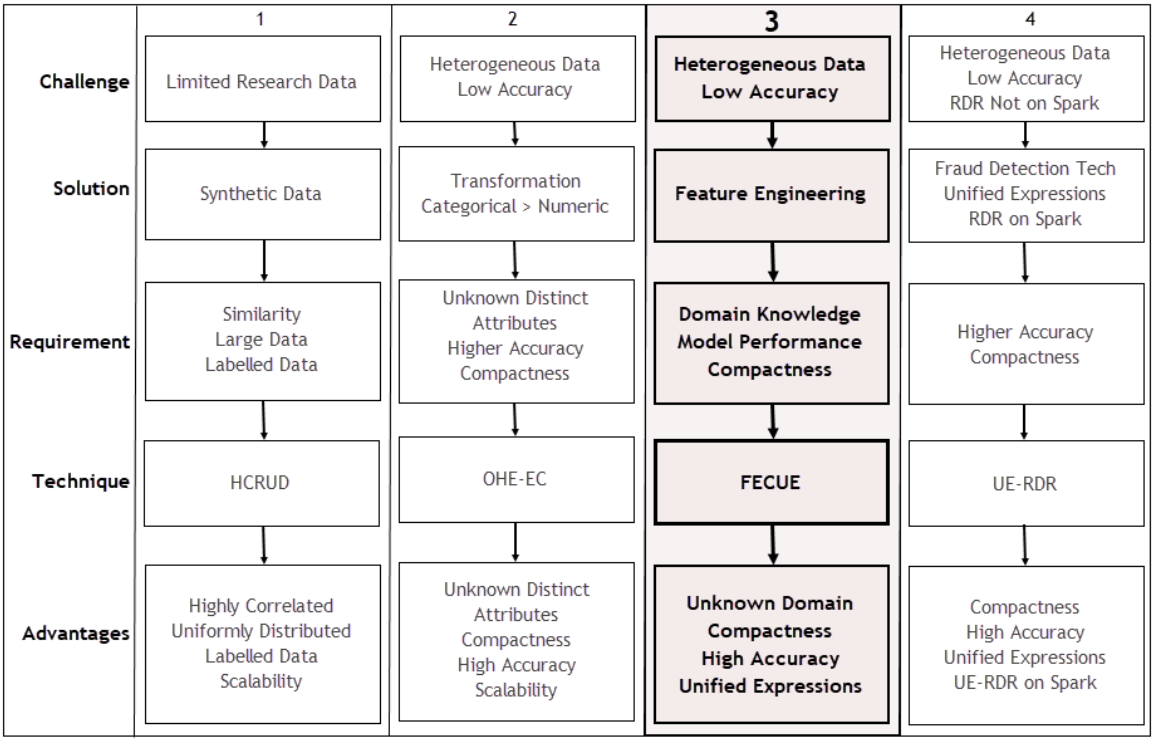
Datasets size of different formats including synthetic data of mixed data and data generated by OHE and OHE-EC were compared. It was observed that datasets size is smallest with OHE-EC, as an average the data in OHE-EC is 3x reduced from OHE. Classification accuracy with OHE-EC with HDF model is also slightly better as compared to the mixed dataset, OHE and OHE-EC (FCFS). Model training and prediction time are also improved with OHE-EC.

4.5 Conclusion

FD for online banking is an important area of research, but the heterogeneous nature of data (i.e. mixed data) is challenging. Numeric format data is known to give better performance with classification and some ML platforms such as Apache Spark by default only accept numeric data. OHE is a widely used approach for transforming categorical features to numerical features, but in various datasets, the distinct values of an attribute are not always known in advance. Also, the sparseness of the transformed data is another challenge. Due to growing memory and storage consumption needs; compactness of ML models has become much more critical. An innovative technique is presented in this chapter to transform categorical features to numeric features by compacting sparse-data even when all the distinct values are not known. Results produced by this technique are demonstrated on synthetic and real bank fraud data and AD KDD-99 datasets on the multi-node Hadoop cluster. The empirical results show that One-hot Encoding Extended (OHE-E) gives improvements over mixed datasets and One-hot Encoding Extended compact (OHE-EC) not only gives a further improvement in reducing the size of datasets but also an improvement in model's training and prediction time. Two further models OHE-EC (FCFS) and OHE-EC (HDF) are also developed in One-hot Encoding Extended Compact (OHE-EC) technique, where OHE-EC (HDF) gives slightly better classification accuracy as compared to OHE-EC (FCFS).

Chapter 5

Enhancing Model Performance for Fraud Detection by FE and Compact UEL



Chapter Overview

The previous chapter (Chapter 4) has presented the overview of categorical feature transformation with Compact One-hot extended (OHE-EC) technique, showing that this could improve system's performance. In the current chapter, we examine how the performance may be further enhanced via FE.

The 3rd highlighted block from the figure describes the part of the research contained in this chapter. It demonstrates the model's performance via FE. This chapter describes the distinct features of an FE technique to improve model performance (FECUE), with no prior knowledge of the domain of the datasets. This chapter also provides an overview of empirical evaluation with classification accuracy using Decision Tree, RDR and Random Forest with multiple datasets. The methodology is further explained with FE on Bank dataset and the performance improvement results on various datasets specified in Table 5.8.

Parts of this chapter were published in: (Ul Haq et al., 2019) Enhancing Model Performance for Fraud Detection by Feature Engineering and Compact Unified Expressions (FECUE), Data Mining: 19th International Conference on Algorithms and Architectures for Parallel Processing, ICA3PP 2019, Melbourne, Australia. This chapter provides an expanded discussion of some aspects of the methodology and datasets, which had to be omitted from the published conference paper due to space restrictions.

5.1 Introduction

The accuracy of an ML model can be boosted with the use of various methods such as segmentation (Bijak & Thomas, 2012), adding more data, treating missing (Xiaofeng et al., 2011) and outlier values, FE (Turner et al., 1999; Xu et al., 2012; Yu et al., 2010) feature selection, multiple algorithms, algorithm tuning and ensemble methods. Particularly, FE helps to extract more information from existing data by deriving new features from existing features. It helps to unleash the hidden relationships in a dataset. Derived features may help in explaining the variance in the training data more accurately and result in higher accuracy. (Bahnsen et al., 2016) also emphasize that while constructing an FD model, it is very important to extract the appropriate features from transaction data. FE could be done using

indicator variables, features interaction, feature representation by extracting information from the existing features, transforming categorical to numeric features, by creating dummy features or by using external data. Feature representation can be mainly applied to categorical attributes. In this chapter, we have focused on feature representation with minimum knowledge of the domain of an external dataset. One of the challenges in FE is to determine if FE can be applied on a particular feature and whether it could be applied via contextual expressions or via external sources, while another challenge is that data become high-dimensional as new features are derived from existing features. We have developed a Feature Engineering and Compact Unified Expressions (FECUE) technique to improve model performance with FE with minimal prior knowledge of the domain of the dataset coupled with compacting the ruleset and dataset with UE using a model-based approach. Performance is measured using three well-known classifiers (Decision Tree (Quinlan, 1992), RDR (Compton & Jansen, 1988; Richards, 2009) and Random Forest (Breiman, 2001). The proposed technique is applied to Bank datasets and two public datasets from UCI ML repository (German Credit and Adult Census Income), explained in Table 5.8. The empirical evaluation has shown that the model's performance has improved while training and prediction model sizes have also been reduced. Main contributions are listed below:

- Study of FE and UE to improve fraud analysis.
- Development of FE technique using custom and configurable SPM when the domain of a dataset is not known in advance.
- Empirical evaluation of the developed technique with multiple datasets.
- Ruleset compactness using contextual expressions and SPMs**.**
- Evaluating performance in terms of standard performance metrics including classification accuracy, precision, recall, f-measure, time and ruleset size.

5.2 Related Work

Some of the known methods of improving model performance are highlighted below:

- Segmentation (Bijak & Thomas, 2012) by dividing the population into several groups.

- Adding more data to produce more accurate models and treating missing (Xiaofeng et al., 2011) and outlier values.
- FE (Turner et al., 1999; Xu et al., 2012; Yu et al., 2010) extracting more information from existing features.
- Feature selection by finding and the most important subset of features.
- Multiple algorithms by applying a relevant model to see better suitability of models for a particular domain.
- Algorithm tuning by finding the optimum parameter values used in the algorithm.

Our research focuses on FE, which is being used in different domains to improve model performance. In (Yu et al., 2010), authors have conducted an educational data mining study; and evaluated FE for KDD Cup 2010 by training the model from students' past behaviour and then predicting future performance. Authors in (Xu et al., 2012) have designed an information extraction technique using FE with a combination of rule-based and ML methods. This technique is applied to narrative clinical discharge summaries. (Turner et al., 1999) have proposed the concepts of FE and have evaluated its impact on the software development life cycle. The authors proposed their research as the first step towards the development of FE and its relationship to other domains. A text classification FE technique is developed by (Garla & Brandt, 2012), which is ontology guided. This technique utilizes the domain knowledge encoded in the taxonomical structure of the Medical Language System with the help of context-dependent relatedness between pairs of concepts.

These developed techniques have a variety of limitations and are either domain or context-specific. They do not discuss the problem or the solution to the increase of data dimension with the application of FE. Also, the impact on the performance in terms of either of the classification accuracy, time and model's size is not discussed. FE via external sources is also not used in these techniques. Considering these limitations, we have proposed an innovative technique which improves model performance over a variety of performance metrics. The proposed technique is an SPM based and domain-independent FE technique using compact unified expressions.

5.3 Methodology

Out of various methods available for improving model accuracy, research in this chapter focuses on FE and compression of ruleset of the training model. One of the challenges was to identify appropriate FE methods for individual attributes, ideally requiring minimal domain knowledge. Another challenge was the compactness of the ruleset. Four SPMs are developed and used in this technique to predict features, which type of FE to use and how to apply the ruleset compactness. SPMs are explained in section 5.3.1. SPMs make the technique more generic for different datasets. Nomenclature of a typical bank transaction log is explained in section 3.3.1 and Table 3.1.

Categorical attributes represent a type of data which may be divided into groups. Typically, a categorical attribute represents discrete values and have no concept of ordering the values of that attribute. From Table 3.1, some of the fields can be used for feature extraction. The developed technique is divided into two parts, feature representation and compactness of the ruleset. An SP (Vastenburg, 2004) defines values relative to the situations, so these are only applied in situations for which they are valid. An SP could help in intelligence extraction efficiently. In RDR, the RDM modelling is also based on SPs (Maruatona, Vamplew, & Dazeley, 2012), as it describes every attribute for a particular case. The developed technique is explained in more detail in section 5.3.4.

5.3.1 Feature Engineering Techniques for Bank Dataset

Many classification algorithms do not use attributes like Event-time, IP Address and Browser string as these types of attributes are ignored in the feature selection process. FE (Ré et al., 2014) is a critical and underexplored aspect of building high-quality KB construction systems and is an understudied problem relative to its importance, especially in FD. One way of FE is extracting information from the existing features, while another way is by using external data sources with some application program interface (APIs) or source like geocoding and demographics. In this chapter, we have also applied FE with external data sources.

If we derive new attributes from existing attributes and train the model, we can see that the new attributes are used by the classifier. The newly derived features either can be numeric or can be easily transformed into numeric attributes. Numeric features give better performance in ML algorithms. Similarly, clustering algorithms work effectively on the data where all attributes are either numeric or categorical data, as compared to mixed data types (Shih et al., 2010). (Ul Haq et al., 2018) also proved higher classification accuracy with numeric data opposed to mixed datasets. In bank dataset, more attributes can be derived from Event-time, e.g. hour, day, month, year, day-of-week, holiday and weekend-flag. Browser string attribute may further produce attributes like O.S, browser and device identifiers. New attributes derived from an IP Address value could be either four segments separated by token character or location-based attributes. External data sources are available which provide geographic information of an IP Address. These newly derived attributes could also be helpful in identifying suspected transactions in terms of fraud. For example, if event hour is not in normal time, or if it is a holiday or weekend or if the location of the IP Address is different from the actual user's location, then there is a higher chance of potential fraud. Same applies with the attributes derived from Browser string attribute. Different SPMs are formed to aid this method to be generic and domain-independent.

5.3.2 Situated Profile Models (SPM)

A Situated Profile (SP) is helpful for efficient extracting of intelligence. RDM model in RDR is Situated Profile based (Maruatona et al., 2012). (Vastenburg, 2004) also highlights that an SP is used between MCRDR engine and the outlier detectors. A number of Situated Profile Models were developed to process features and for the ruleset compactness. These models are used for banking dataset but could also be modified for a specific dataset. Table 5.1 SPM is a set of tokenizer characters and their applicability to attributes, while Table 5.2 explains different measures to predict an attribute based on the type and category. While Table 5.3 FE could be categorized if it can be done via contextual expressions e.g. extracting day-of-week from date field or getting geocoding and demographic information from an IP Address.

Table 5.1: Tokenizer Character Model Sample

Token Character	Category	Attribute Index
.	Include	2, 6
_	Include	3, 5, 4
;	Include	5
,	Skip	all
)	Skip	5

Table 5.2: Feature Prediction Model Sample

Type	Category	Possible values
Attribute Data Type	Comparison	String, Date, Amount, Integer
Tokenizer	Boolean Exists	Yes/No
Tokenizer	Find	Ref: Table 5.1
Tokenizer	Count	1,2,3
Attribute	Length	0-100

Table 5.3: FE Type Model Sample

FE Source	Attribute Index
Contextual Expressions	3
Contextual Expressions	4
Contextual Expressions	5

Table 5.4 is a sample list of UEL operators, which can be replaced with a simple mathematical operator to achieve compactness in UEL ruleset.

Table 5.4: Rules Compression Model Sample

UEL Operator	Simple Operator	Types
Between	>=	Integer, Amount
Between	<=	Integer, Amount
Like/In	=	String
Not Between	NA	Integer, Amount
Not In	NA	String

5.3.3 Challenges and Tokenizing a Feature Value

One of the challenges in FE is how to evaluate which information or features could be extracted from a particular feature, which already exists in the dataset. It cannot be done without domain knowledge or at-least heuristic approach needs to be applied based on the data type. Without domain knowledge of fraud dataset, how we will know that browser OSVer, O.S, Ver and device features can be extracted from raw Browser string. Heuristically, we know that hour, day, month, day-of-week, holiday and weekday flag information can be extracted from a date-time feature and that an IP Address contains geolocation data, which can be extracted by some external APIs.

A new way of FE is introduced in this chapter, which can extract information from existing features with minimum domain knowledge of the dataset. Four SPMs (Table 5.1 – Table 5.4) are developed in this technique to predict a feature and to decide the source of FE. The technique is explained in Algorithm 5.1 and in section 5.3.9 with a rule-based approach. By using this algorithm and the suggested rule-based approach, information can be extracted by tokenizing a feature value with non-alphanumeric characters, e.g. comma, space, bracket, colon and semicolon, Table 5.1 is configurable to update tokenizer characters with respect to attributes. From a sample date-time value “15/10/2018 23:55:10” six numeric attributes can be extracted by using Algorithm 5.1, which are “15 10 2018 23 55 10”. A classifier doesn’t need to know which value an hour is, day, month or a year. Similarly, from a sample Browser string value “Mozilla/5.0 (iPad; CPU OS 3_2_1 like Mac OS X; en-us) AppleWebKit/531.21 (KHTML, like Gecko) Mobile", O.S, browser and device identifiers can be extracted. Although the contents of a Browser string will slightly vary based on the browser and the underlying operating system, once the system knows that it is a Browser string field, it can further extract these attributes. A ruleset can be further developed to extract browser name, operating system and the versions, as Browser string contents may vary based on the browser and the O.S. These newly extracted attributes are a combination of categorical and numeric attributes. But the extracted categorical attribute can also be converted to numeric attribute, which was not possible with the original attribute value of Browser string. Various SPMs are developed in this technique for Bank dataset but may also be customized for a particular dataset.

In these profile models, tokenizers' base can also be built, for example for a particular browser string value "like" could also be a tokenizer/split string. If we also use "_" as tokenizer, then we can also extract feature from "Source Account" as "Home_Loan" and "Personal_Loan". Table 5.5 shows derived attributes when FE is applied to a Source Account field in banking dataset.

Table 5.5: FE Applied on Source Account

Original Field	**Derived Fields**	
Source Account	Source Account	Source Account
Personal	Personal	Personal
Personal_Loan	Personal	Personal_Loan
Home_Loan	Home	Home_Loan

5.3.4 Algorithms

The developed technique is based on FE and compactness of ruleset for the model. FE is explained in Algorithm 5.1, while ruleset compactness is explained in Algorithm 5.2. Tokenizer characters are maintained in SPs for every attribute, as a particular character could be a tokenizer character for one attribute, but not valid for other attributes.

5.3.4.1 Algorithm 5.1

#Load Source data and perform data cleaning.

#Do feature selection and filter categorical features and other features having tokenizer characters.

ALGORITHM 5.1: Feature Engineering

Input: Instance from a dataset.

Output: Instance with the addition of new features with FE.

Begin

1. Process instances.
2. Process each Feature
3. IF Feature (Is Categorical) or (Having tokenizer characters)

i. Categorise the feature based on Table 5.1 and Table 5.2 (explained in more detail in section 5.3.9)

ii. For each feature transform and extract new features with FE.

iii. Tokenize / Split with Tokenizer characters from SPs using Table 5.1 and Table 5.2

FOR Feature 1 to n LOOP

IF NEW Tokenizer THEN Update SPs

SPs will manage collection of tokenizer characters on attribute level.

ELSE IF Tokenizer THEN NewFeatures = fExtractFeatures(feature)

#Extract feature with the token

NEXTVALUE

ENDLOOP

4. IF (more features in the row) Goto Step 2

#Extract features from complete Row from Step 2 - 4, IF (more Row) Goto Step 1

ELSE FINISH

End

5.3.4.2 Algorithm 5.2

ALGORITHM 5.2: Compactness

Input: A unified expression format rule from a ruleset.

Output: A compact unified expression format rule.

Begin

#Load Ruleset.

1. Process each rule in the ruleset and compact the ruleset using funcCompact function Eq. 5.1.

2. Process each expression in the rule.

3. IF (Expression is >= or <=) Process current rule and update UEL Rule 3.a

#Update UEL Rule with BETWEEN operator

ELSE if (Expression is ==)

#Process current rule and update UEL Rule 3.a. Update UEL Rule with UEL operators as Table 5.4

ELSE SKIP

ENDIF

3.a Update Unified Expression Rule (UEL)

#Update with appropriate UEL operator (BETWEEN, IN, NOT IN, LIKE, NOT LIKE) as explained in Table 5.4 and in section 5.3.6

4.IF (more expression) Goto Step 2

#Process expressions from complete Rule from Step 2 - 4. IF (more Rules) Goto Step 1

ELSE FINISH

End

5.3.5 Feature Engineering for Bank Dataset

A sample of records from the Bank dataset is shown in Table 5.6, while Table 5.7 shows the same data sample after FE. Figure 5.1 shows a RIDOR ruleset generated from Table 5.7 dataset.

Table 5.6: Bank Dataset (Original)

Acc Type	Source Acc	Event Time	Browser String	IP Address	Class
FT	Credit	13/12/17 1:12	Alt webkit Unk/Unk x64	14.44.27.11	None
PA	Personal	18/04/17 9:58	Alt webkit Unk/Unk x64	16.19.13.16	None
FT	Personal	24/07/17 4:31	Alt webkit Unk/Unk x64	73.17.22.19	Fraud
PA	Home_Loan	8/09/17 3:46	Alt webkit Unk/Unk x64	15.55.24.11	None
FT	Personal	19/02/17 8:45	Alt webkit Unk/Unk x64	99.22.21.15	None
PA	Personal	9/08/17 2:46	Moz_5 webkit Win/Lap x64	18.15.92.11	None
FT	Personal	20/09/17 4:07	Moz_4 webkit Unk/Mob x64	99.12.21.54	None
BPAY	Personal	21/10/17 1:38	Moz_4 webkit Unk/Mob x64	18.19.20.10	None

Table 5.7: Bank Dataset (with Derived Attributes)

Acc Type	Source Acc	Hour	Day	Month	DOW	Week day	Browser VerOS	Browser OS	Browser Ver	Device	State	City	Country
FT	Cre	1	13	12	wed	wday	Alt	Unk	0	Unk	Oth	Oth	Oth
PA	Pers	9	18	4	wed	wday	Alt	Unk	0	Unk	Oth	Oth	Oth
PA	Bus	11	4	8	sat	wend	Moz 4	Win	4	Mob	CL	Sand	US
PA	Pers	3	26	9	tue	wday	Moz 5Win	Win	5	Lap	NSW	Sydney	AU
PA	Pers	0	20	10	sun	wend	Moz_4	Unk	4	Mob	VIC	Melb	AU
FT	Pers	18	19	2	fri	wday	Alt	Unk	0	Unk	Oth	Other	Oth

5.3.6 Unified Expression Language (UEL)

In this chapter, we have considered rule-based classifiers. One of the well-known classifiers is RDR. We have suggested ruleset compactness in RDR using UE using SPMs. UEL can evaluate mathematical expressions with a lot of operators and enables dynamic scripting feature. Some of the advantages of UEL is that it supports more than 30 different operators; and expressions can also invoke functions, which can help in getting external data for FE. For example, extracting geolocation data in Bank dataset. Rule-based classifiers use only limited operators. However, using UEL many more operators can be used e.g. IN and LIKE

operators. In FE, features interaction can be achieved by dynamically evaluating expressions using Add, Subtract, Multiply and Divide operators instead of creating new features in the prediction phase. FE with feature interaction will be only needed for training the model. Authors in (Ul Haq et al., 2018) have highlighted the importance of compactness of the prediction model and demonstrated that a compact prediction model is more efficient. The UEL expression will help in ruleset compactness and will improve performance in terms of the time taken for model prediction.

Algorithm 5.2 explains compactness with UEL using a configurable SPM Table 5.4. This model uses a relevant UEL operator which can be used based on simple operator and attribute type. Ruleset compactness with UE is explained below:

Rule-1: 'Source_Acc'='Personal' and 'Country'='AU' and Browser=’MOZ-5Win’ THEN FRAUD

Rule-2: 'Source_Acc'='Personal' and 'Country'='AU' and Browser=’MOZ-5Lin’ THEN FRAUD

Compressed Rule: **(**Using IN Operator**)**

'Source_Acc'='Personal' and 'Country'='AU' and Browser IN (’MOZ-5Lin’, ’MOZ-5Win’) THEN FRAUD

Other Operator could be BETWEEN for numeric features and LIKE for categorical features.

The compactness of expression is explained with Eq. 5.1.

$$R^{comp} = funcCompact \int_{i}^{n_Y} (expSet)\ m \neq \text{null} \quad (5.1)$$

where expSet is a set of expressions from RDR ruleset and R^{comp} is a compact ruleset with UE and funcCompact is a function to compact an RDR ruleset which compacts simple mathematical expressions from 1 to n from SPM Table 5.4 on i^{th} rule index having m value which is non-null.

5.3.7 Ripple Down Rules Ruleset

RIDOR is most widely used RDR machine learner, while J48 is Decision Tree implementation in Weka ML tool. RDR classifier is used for the dataset with derived attributes Table 5.7. The ruleset generated by RIDOR classifier is listed in Figure 5.1, confirming that the newly extracted features are used in the training model.

```
Class = Anon (1756.0/1583.0)
Except (Browser = Alt) => Class = Non (528.0/0.0) [270.0/0.0]
Except (Network_Count > 11) and (Network_Count <= 12.5) => Class = Fraud (37.0/0.0) [18.0/0.0]
Except (Country = AU) => Class = Fraud (425.0/0.0) [214.0/0.0]
Except (UserCity = Sydney) and (Login_Count > 3.5) => Class = Non (137.0/5.0) [56.0/5.0]
Except (UserState = ACT) and (Hour > 9.5) and (Source_Amt <= 7990) => Class = Non (10/0) [8/3]
Except (UserState = QLD) and (UserCity = Brisbane) and (PwdChange > 0.5) => Class = Non (9/0) [2/0]
Except (UserCity = Perth) and (Weekday_flag = WEEKEND) => Class = Non (6.0/0.0) [2.0/0.0]
Except (Browser = Moz_5Win) and (PwdChange <= 0.5) and (Month <= 4.5) => Class = Fraud (6/1) [5/2]
Except (IPCity = Sydney) and (Source_Amt <= 2583.5) => Class = Non (6.0/0.0) [1.0/0.0]
Except (Country = CZ) => Class = Fraud (3.0/0.0) [2.0/0.0]
Except (Country = US) => Class = Non (2.0/0.0) [2.0/0.0]
```

Figure 5.1: Ripple Down Rules Classifier Ruleset

5.3.8 Contextual Expressions

UE can be used to get further useful information from the existing attributes through external sources, e.g. getting geocoding and demographic information from IP Address in Bank dataset. Which can help in making further decisions related to fraudulent transactions and will improve model accuracy as well. To make it generic which attributes needs FE from an external source, an SPM Table 5.3 is developed and used in this technique. This model decides the FE based on the attributes, which is predicted from two other models given in Table 5.1 and Table 5.2. e.g. get country information from IP Address may help in detecting suspected tunnel sites usage. We can add a rule when IP Address and user's actual country are different.

Rule: 'Source_Acc' == 'Personal' and 'UserCountry' <> 'IPCountry' THEN FRAUD

5.3.9 Constructing a Feature

Extracting features from the existing feature is a challenging task, especially without knowing the domain of the dataset. However, if we know the feature name in a particular dataset, it will help in extracting more features from this feature. Considering commonly used data types explained by (Durrant et al., 2018; Witten, Frank, Hall, & Pal, 2011) and adding some further measures of feature content length and presence of the token character, a rule-based approach is developed to predict a feature name. To make the technique more generic, four SPMs are developed and used in this technique. See a ruleset example:

Rule-1: DataType=’String’ and Count (Token_Character=’.’) = 3 THEN IPAddress

Rule-2: DataType =’String’ and Token_Character==’;’ THEN BrowserString

Rule-3: DataType =’String’ and (No_Token_Character or Token_Character=’_’) THEN SourceAccount

Rule-1, 2 and 3 can also be represented as:

DataType =’String’

Count (Token_Character=’.’) = 3 THEN IPAddress

Token_Character=’;’ THEN BrowserString

(No_Token_Character or Token_Character=’_’) THEN SourceAccount

Comparison with attribute types and checking the existence of a particular attribute and using other measures of length or count from the SPMs, which is explained in section 5.3.1.

5.4 Results

The empirical evaluation was done for both original and the dataset produced by FECUE technique. The performance was measured with a variety of performance metrics including classification accuracy, precision, recall, f-measure, time and ruleset compactness.

$$\text{Accuracy} = \frac{TP+TN}{\text{TP+FP+FN+TN}} \tag{5.2}$$

$$\text{Precision} = \frac{TP}{\text{TP+FP}} \tag{5.3}$$

$$\text{Recall} = \frac{TP}{\text{TP+FN}} \tag{5.4}$$

$$\text{F-measure} = \frac{2*(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \tag{5.5}$$

where TP are correctly predicted positive and TN are correctly predicted negative values, FP when actual class is no and predicted class is yes and FN when actual class is yes but predicted class is no. Along with many performance measures for classification, accuracy, precision, recall and f-measure are well explained by (Hackeling, 2014).

5.4.1 Dataset Characteristics

Characteristics of multiple datasets used for the evaluation are explained in Table 5.8. This table also shows the number of additional features that FE has added to each dataset.

Table 5.8: Data Characteristics

Dataset	Instances	Features	Features addition with FE
Reference Bank data (Ul Haq et al., 2016)	1,756	14	9
Synthetic Bank data (Ul Haq et al., 2016)	50,000	14	9
German Credit data(Hofmann, 1994; Prasad & Ramakrishna, 2016)	1,000	11	4
Adult (Census Income) data (Kao, Chung, Sun, & Lin, 2004; Zadrozny, 2004)	32,562	8	6

Synthetic Bank data was generated from reference Bank data using HCRUD (Ul Haq et al., 2016) technique, where class labels and attributes in the generated data were evenly distributed as in original reference data.

5.4.2 Bank Datasets

Various performance metrics with three well-known classifiers have been compared with the use of the original datasets and corresponding datasets, with derived attributes after FE using FECUE. The results in Table 5.9 and Table 5.10 show that there is an improvement in the performance metric. In this study, 30% and 70% split is done for training and testing datasets. Average measurement was calculated for various dataset sizes ranging from small to large datasets and for multiple simulation runs for each classifier.

Table 5.9: Performance with Reference Bank Dataset

Classifier	Accuracy	Precision	Recall	F-measure	Time	Ruleset
RIDOR	3.96%	1.85%	4.05%	4.05%	58.06%	26.09%
C45/J48	0.32%	-0.10%	0.00%	0.00%	50.00%	-10.67%
Random Forest	49.39%	91.49%	33.68%	97.39%	-8.33%	

Table 5.10: Performance with Synthetic Bank Dataset

Classifier	Accuracy	Precision	Recall	F-measure	Time	Ruleset
RIDOR	6.75%	7.34%	6.75%	7.91%	165.32%	50.32%
C45/J48	2.64%	5.87%	6.37%	2.53%	108.41%	15.53%
Random Forest	50.58%	52.42%	50.58%	119.64%	20.26%	

Above tables show that there is an overall improvement (original and corresponding datasets after FE with FECUE) in all performance metrics with both bank's datasets.

5.4.3 Public Datasets

FE was also applied on two publicly available datasets: German Credit data (Hofmann, 1994; Prasad & Ramakrishna, 2016) and an Adult (Census Income) (Kao et al., 2004; Zadrozny, 2004) dataset. Table 5.11 and Table 5.12 show the results and depict that there is also improvement in the performance metric results for three classifiers. Results have shown that classification accuracy is also improved, but there is slightly lower accuracy improvement as compared to the Bank datasets. Reason for low improvement is that fewer new attributes were added in public datasets as compared to the Bank datasets. Reference Bank dataset and German Credit dataset are very small as compared to the other two datasets. The only ruleset for Decision Tree and time performance metric for Random Forest in reference Bank data are slightly degraded, the actual difference is very small but it is showing in improvement in percentage. But there is an overall improvement in other performance metrics, which is shown in Tables 5.11 and Tables 5.12.

Table 5.11: Performance with German Credit Dataset

Classifier	Accuracy	Precision	Recall	F-measure	Time	Ruleset
RIDOR	0.93%	0.29%	0.97%	-0.58%	23.53%	20.00%
C45/J48	2.34%	2.53%	2.38%	2.39%	-3.03%	1.52%
Random Forest	5.45%	5.01%	5.46%	4.56%	35.00%	

Table 5.12: Performance with Adult (Census Income) Dataset

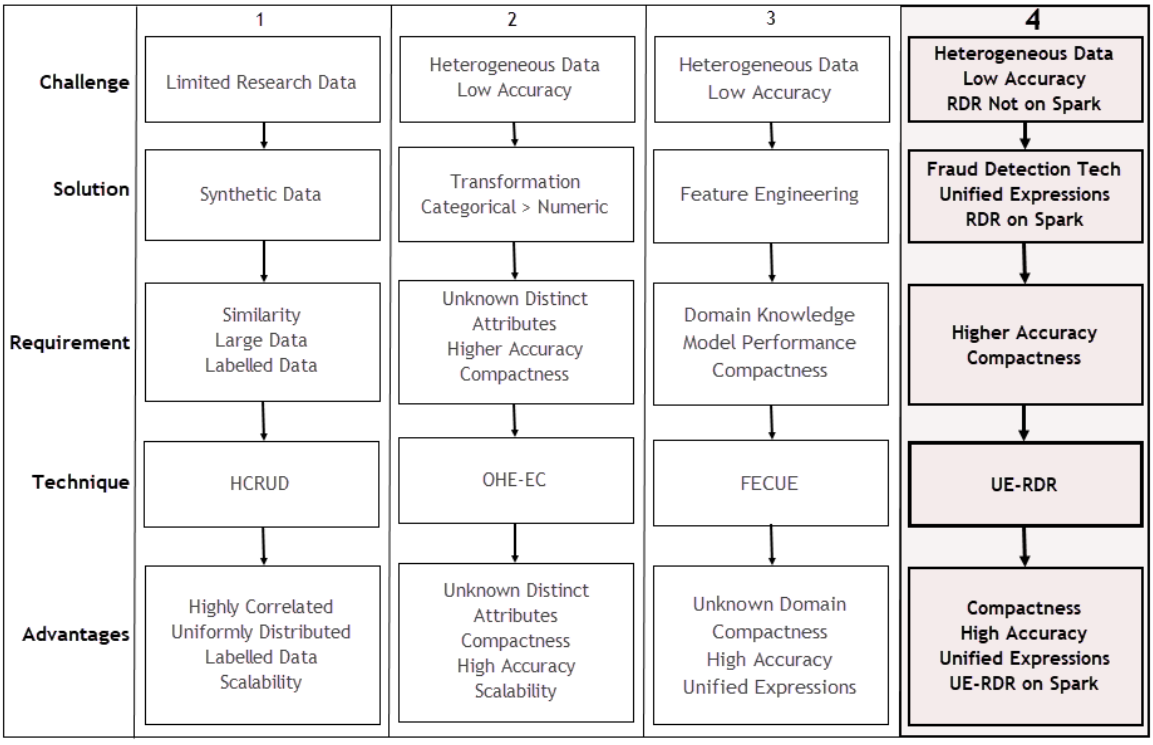
Classifier	Accuracy	Precision	Recall	F-measure	Time	Ruleset
RIDOR	1.53%	0.60%	6.54%	4.11%	35.78%	4.55%
C45/J48	1.28%	1.06%	1.30%	1.42%	32.70%	42.00%
Random Forest	1.29%	1.53%	0.12%	1.29%	12.40%	

5.5 Conclusion

Model performance can be improved in a variety of ways including segmentation, treating missing and outlier values, FE, feature selection, multiple algorithms, algorithm tuning and ensemble methods. This chapter has presented model accuracy and compactness technique (FECUE), and it is observed that the derivation of new features makes the dataset high-dimensional. The developed technique has enhanced the model's performance with FE (when the domain of a dataset is not known in advance), with the use of external sources and compact, UE. Multiple SPMs are used to make the technique more generic so that it is applicable to multiple datasets and domains. Performance in terms of classification accuracy, precision, recall, f-measure, time and ruleset compactness is compared using three well-known classifiers. FECUE has been applied on reference bank, multiple synthetic bank and two publicly available datasets: German Credit and Adult (Census Income) datasets. The empirical evaluation has shown that not only the ruleset in training and prediction model are reduced, but the performance improvement is also observed in other standard performance metrics. The developed technique is mainly applied in the FD area, but it can be used in other domains as well. One of the future works would be to test this technique on a variety of datasets, especially with high-dimensional data.

Chapter 6

Unified Expression Ripple Down Rules based Fraud Detection Technique for Scalable Data



Chapter Overview

The previous chapter (Chapter 5) has presented an overview of FE technique to improve model performance. In the current chapter, high classification accuracy challenge on mixed datasets, especially for scalable data in RDR is addressed. It also addresses the RDR implementation challenge on Spark platform. The following Chapter 7 concludes the whole thesis, indicating the limitations and possible future research.

The role of the work in this chapter within the overall research is explained by the last highlighted block from the figure. It demonstrates the research problem and the technique developed. This chapter describes the features of the developed technique. It explains Unified Expression RDR Fraud detection technique (UE-RDR) for scalable and distributed data. The chapter gives an overview of the three models developed in this technique and also provides an overview of empirical evaluation and comparison with two RDR based classifiers (RIDOR and IPA) and Naïve Bayes (a non-RDR classifier) with multiple datasets.

The work in this chapter has been submitted for reviews as (Ul Haq et al., 2020). Unified Expression Ripple Down Rules based Fraud Detection Technique for Scalable Data, Data Mining: Australasian Information Security Conference, AISC 2019, Melbourne, VIC, Australia.

6.1 Introduction

FD for online banking is vital as frauds can affect the core business of the financial industry in terms of loss of confidence of the public in the industry. IC3 has reported a 161% increase in the loses in 2018 (FBI, 2018). Various FD techniques have been developed over the last decade. In view of the importance of FD in the banking sector, higher accuracy of FD techniques is critical. One of the major challenges faced by fraud analysis research is the heterogeneous nature of transactions (Ul Haq et al., 2018). Typically, datasets can have both numeric and alphabetical attributes, but numeric data is known to provide better performance for ML algorithms. Large-scale data in online banking also requires algorithms to show better performance with scalable and distributed data. (Meng et al.,

2016) highlight that Apache Spark is a popular open-source platform for large-scale data processing and iterative ML tasks.

6.1.1 Prior Work on Fraud Detection Using Machine Learning

(Kou et al., 2004) believe that FD research mostly uses data mining, statistics, and artificial intelligence; and fraud is identified from anomalies in data and patterns. (Phua et al., 2010) have surveyed FD research to categorize the research using four main approaches including supervised, hybrid, semi-supervised and unsupervised and; also identified the relationship of FD with other domains. (Melo-Acosta et al., 2017) have presented a credit card FD technique using Big data, but their technique is more specific to imbalance and unlabelled data.

(Herland et al., 2018) have presented an FD approach for Medicare fraud using three medicare and medicaid services datasets. The authors use the combined dataset for training with three learning methods: Random Forest, Gradient Tree Boosting and Logistic Regression models and used the Area Under Curve (ROC) metric to measure the performance of FD. They claim that best FD performance is with the use of the combined dataset. Dataset size is not mentioned, but this technique is not ideal for large datasets, e.g. Synthetic dataset generation based on original seed datasets.

IPA is developed by (Maruatona, 2013) which uses PA in RDR and has combined two of the previously developed Multiple Classification RDR (RM) and RDM (Kang et al., 1995; Prayote, 2007) techniques. A fundamental difference in these techniques is that RM is structural while RDM is attribute-based. The difference in these methods is well explained by (Maruatona et al., 2012). IPA is a multi-class labels classifier.

6.1.2 Background to UE-RDR Methodology

RDR is one of the well-known rule-based classification technique and is developed as an alternative to the traditional KBS (Compton & Jansen, 1988; Kang et al., 1995). (Richards, 2009) acknowledges that RDR is ideal due to its less maintenance and incremental learning capabilities. RDR significantly reduces the time and effort required to make the alteration and ensure the consistency of the rulesets. (Kang et al., 1995; Richards, 2003) have

highlighted that RDR systems have been used in many applications and classification domains. RIDOR is an RDR implementation in Weka and (Compton, 2011) also acknowledges that RIDOR is most widely used RDR machine learner. Iris is a small dataset (Appendix A), used to generate a sample ruleset. Figure 6.1 shows the ruleset generated for Iris Dataset from RIDOR.

```
class = setosa (150.0/100.0)
Except (petal_len > 2.45) => class = virginica (66.0/0.0) [34.0/0.0]
Except (petal_len <= 4.95) and (petal_wid <= 1.55) => class = versicolor (29.0/0.0) [16.0/0.0]
Except (petal_wid <= 1.75) => class = versicolor (8.0/5.0) [1.0/0.0]
```

Figure 6.1: Iris RIDOR Ruleset

One of RDR implementation is RIDOR, which also has MapReduce (ASF, 2015) based implementation in Weka for Apache Hadoop (ASF, 2015) wrapper, which can be used for the classification of large data. However, (Meng et al., 2016; Shanahan & Dai, 2015) highlight that Spark is better as compared to conventional MapReduce. Spark maintains MapReduce's linear scalability and fault tolerance and is nearly 100 times more efficient than MapReduce. Mahout is another ML platform for Big data. (Meng et al., 2016) highlights that Mahout is also based on MapReduce and they observed that Spark's performance and scalability are better than Mahout.

UEL is a special language for embedding expressions. UEL is capable of evaluating a number of additional operators that are missing in RDR model's ruleset expressions. Unified Expressions (UE) in UEL can also replace existing operators with more efficient operators of IN and LIKE. Using UE, we can prepare compressed rule with a revised Lift score which is the ratio of target response divided by the average response. UEL supports contextual expressions and can also retrieve geocoding and demographics information from fraud datasets (Ul Haq et al., 2019), that helps to filter suspected cases. UE application in the proposed technique is explained in section 6.2.6. UE can offer a variety of operators that can help with the compactness of ruleset and evaluation of the expression based on Lift

score. Furthermore, the UE can help in choosing the best rules with higher confidence; therefore, the more accurate class label is chosen, which improves accuracy. UE-RDR is implemented on Big data Spark platform by overcoming the limitation of mixed datasets. Apache Spark performance is known to be better than conventional Apache Hadoop MapReduce (Meng et al., 2016; Shanahan & Dai, 2015) so UE-RDR on Spark will be more efficient than RDR MapReduce based implementation in Weka and will also have iterative ML capability.

UE-RDR FD technique for large-scale mixed data has been developed and evaluated in this chapter to improve detection accuracy and reduce computation costs. The technique has three models: the minority (UE-RDR-MIN) class, the majority (UE-RDR-MAJ) class-based models and combined model (UE-RDR-MIX). The combined and distinct rules in UE-RDR-MIX model gives better accuracy than the other two models. UE-RDR-MIX is an innovative model and to the best of our knowledge, no study has been on in RDR based classifiers. UE-RDR performance is compared with RDR. The proposed technique is applied to various data datasets (Table 6.3), including Synthetic Bank datasets and three publicly available datasets from the UCI ML repository. Performance is evaluated and compared with two RDR based implementations (RIDOR and IPA) and a non-RDR classifier (Naïve Bayes (Swain & Sarangi, 2013) as well. The empirical evaluation has shown that the model's performance in terms of classification accuracy and ruleset size is better than RIDOR. Classification accuracy with UE-RDR-MIX is better than IPA and Naïve Bayes classifiers.

The main contributions of the chapter are listed below:

- Study of UE for RDR and development of a threshold-based approach for ruleset compression with the use of Lift score.
- Development of a single classification Unified Expressions RDR (UE-RDR) technique with three sub-models: UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX. UE-RDR-MIX is an innovative model for RIDOR, which makes use of majority and minority classes and multi-level compactness.

- Empirical evaluation of the developed technique for classification accuracy and ruleset compactness with multiple datasets and comparison with various RDR and non-RDR based classifiers.
- Study of the developed technique on distributed and Big data ML platform, Spark.

In this chapter, we are focusing on FD for large-scale data and with rule-based classifiers using a supervised approach on labelled datasets. The developed technique can be used on mixed datasets. The developed algorithm is implemented on big and distributed data platform Spark and has shown better accuracy as compared with two of the existing RDR based classifiers and a non-RDR classifier.

6.2 Methodology

KBS are a major application for concept descriptions. (Littin, 1996) mentions that rules and Decision Tree are two of the common forms of concept descriptions in ML. (Capterra, 2019; Maruatona, 2013) point out that most of the Internet banking FD systems are using rules-based approaches. He points out that commercial banks and financial institutions are using rules-based approaches.

6.2.1 UE-RDR Models

FD data is a single classification data, and UE-RDR is also a single classification model, with UE based on RDR. In UE-RDR technique, three models are developed, UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX. (Littin, 1996) highlights that the inclusion of RDR top-level empty rule is used generally with a default class. (Gaines & Compton, 1995) have used the class that occurs most frequently (Majority) as default in the training data, however in RIDOR by default least frequently used class (Minority) is used as default class. UE-RDR technique is also illustrated graphically as a multi-step process in Figure 6.2. Figure 6.3 shows iris ruleset for a UE-RDR-MIN model. But a typical ruleset and a particular rule structure of UE-RDR model is shown below:

{"defaultclass":"CLASS-LABEL", "model":"MODEL-NAME","count":TOTAL-POPULATION,

"rulses":[RULES-COLLECTION]

RULE#

{"number":#,"isParent":true,"level":#,"description":"UE-EXPRESSION","lift":#,"cover":#,"ok":#
"class":"CLASS-LABEL","parentid":#, "childrenNodes":#}

6.2.1.1 UE-RDR-MIN

In this model, least frequently occurring (Minority) class is the default class (like RIDOR), and the rules are for the remaining class labels, i.e. majority class label and other classes. In most of the cases ruleset set for this model is supposed to be larger than the ruleset for UE-RDR-MAJ, as least frequently used class is default class and rules are for the remaining class labels (including majority class).

6.2.1.2 UE-RDR-MAJ

In this model, most frequently occurring (Majority) class is the default class (as used by (Gaines & Compton, 1995)) and the rules are for the remaining classes. In terms of ruleset size, this model would have a similar size of ruleset as UE-RDR-MIN model.

6.2.1.3 UE-RDR-MIX

This model is a union of the rules for the minority & majority class models and distinct rules for the remaining class-labels. Rules expressions are further compressed with revised Lift score outlined in sections 6.2.5 and 6.2.7. This model is our innovation and does not exist in RIDOR implementation. Algorithms 6.2a explains this model. In RDR ruleset, one class is the default class and ruleset contain rules for the remaining class labels. We claim that this model gives the best classification accuracy, as shown in Figure 6.5. Unlike RDR, it contains rules for all class-labels instead of using a default-class. In terms of ruleset compactness, Figure 6.6 shows that for some datasets, UE-RDR-MIN and UE-RDR-MAJ have good performance as well.

If there are more than two class labels in a dataset, this model also provides better accuracy for class labels that belong to neither majority nor minority classes. Considering Bank dataset example, the Fraud class label does not fall into the majority or minority class, so UE-RDR-MIX model will give better accuracy for Fraud class labels in this dataset. Apart

from the overall higher classification accuracy, classification accuracy is also sometimes important for a specific class label. For example, Fraud cases are more important for improved accuracy in the Bank dataset. A wrong prediction of a Fraud case would result in a greater loss compared to the mistake of None or Anon cases. Accuracy results from the confusion matrix are shown in Figure 6.8.

6.2.2 Algorithms

The developed technique is based on three algorithms. UE-RDR ruleset construction is explained in Algorithm 6.1, while ruleset compactness is explained in Algorithm 6.2 and prediction flow with Spark is explained in Algorithm 6.3. Algorithm 6.2a is for UE-RDR-MIX model only, which is further compactness of Majority and Minority class models (UE-RDR-MIN and UE-RDR-MAJ). Figure 6.2 illustrates UE-RDR process flow and glues three algorithms to demonstrate the three-stages. In Algorithm 6.3, when a data file is stored in HDFS (ASF, 2015), the system breaks it down into individual blocks set and stores these blocks in multiple worker nodes in the cluster. Rows division in each data block can be determined with Eq. 6.1.

$$Rows^{Block} = \Sigma Rows/SparkNodes/BlockSize/RowDataSize \quad (6.1)$$

The mentioned algorithms are given below:

ALGORITHM 6.1: Building Training Model

Input: Ruleset from a RIDOR.

Output: Training model for a UE-RDR.

Begin

1. Process RIDOR ruleset.
2. Process each expression in the ruleset.
3. Get Ok and Cover values of each expression.
4. Calculate Lift score of the expression from Ok and Cover values using Eq. 6.4.
5. Prepare the expression in UE format using funcUEL Eq. 6.5.
6. Convert the expression in JSON format with attributes (See Figure 6.3).
7. IF (more expressions in the ruleset) Goto Step 2

ELSE FINISH

End

ALGORITHM 6.2: Compactness

Input: Training model for a UE-RDR.

Output: Compact UE-RDR Training model.

Begin

1. Process each rule in the ruleset of the training model.

2. Traverse Ruleset & Get Lift score of the rule

2.1. Find merging rule (using the custom thresholds approach listed in Table 6.2).

2.2. Merge UE rule.

3. Traverse rule to compact UE (See UE operators Table 6.2)

3.1. Calculate and update revised Lift score, from updated Ok and Cover values of merging rule – see Eq. 6.4.

3.1 Update UE rule.

3.2 IF (more expressions to process) Goto Step 3

3.3 Process all expressions from complete rule from Step 3 – 3.2

4 IF (more rules) Goto Step 1 ELSE FINISH

End

ALGORITHM 6.2a: UE-RDR-MIX Compactness

Input: Training model for a UE-RDR-MIN and UE-RDR-MAJ.

Output: Compact UE-RDR Training model for UE-RDR-MIX.

Begin

1. Repeat Algorithm 6.2 with the input of two UE-RDR Training Models.

2. Repeat Steps 1 to 3.2 from Algorithm 6.2.

End

ALGORITHM 6.3: Prediction Process

Input: Training model from a UE-RDR and dataset.

Output: Classification accuracy for the dataset.

Begin

1. Load Dataset

1.1. Process each instance.

1.2. Transform instance to Resilient Distributed Dataset (RDD) double Vector, including categorical attributes using funcTransRDD Eq. 6.2.

1.3. Split data on Spark nodes based on the data block size using Eq. 6.1.

2. Load UE-RDR training model.

3. Load RDD vector collection from data locality.

3.1. Process each rule from the Training Model.

3.2. Transform categorical attributes in expression with funcTransCat function Eq. 6.3.

3.3. Evaluate UE rule expression and pick the predicted class.

3.4. If multiple rules are true, then pick predicted class of better Lift score rule.

3.5. IF (more rules in the ruleset) Goto Step 3.1

IF (more instances to process) Goto Step 3 ELSE FINISH

End

6.2.2.1 UE-RDR Process Flow

Figure 6.2 connects three algorithms to illustrate the flow of the three-step algorithms. The dependency in each step and the main and subtasks in each step are clarified there. Loading and Prediction are the two steps in the Prediction process.

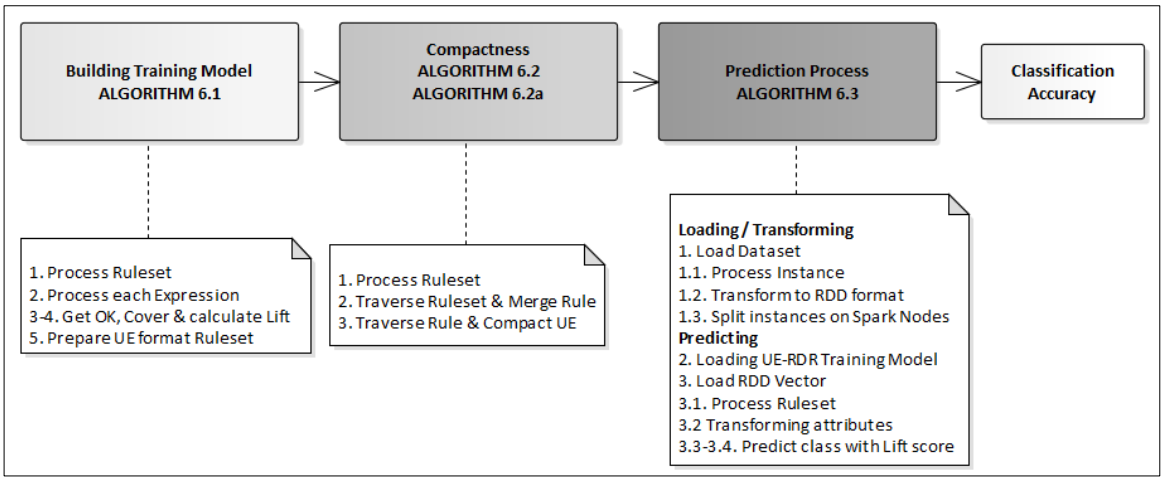


Figure 6.2: UE-RDR Process Flow

6.2.3 Transformations

Due to the large datasets, the developed technique was implemented on Spark. The core of Spark is a concept called the Resilient Distributed Dataset (RDD), which is a collection of records. The default data-format for Spark platform is numeric, however the Bank dataset and many real-life datasets contain mixed attributes. Two transformation functions were developed, which are explained below. The function in Eq. 6.2 transforms mixed data to numeric RDD format at loading time.

$$\text{Transformation}^{\text{RDD}}=\text{funcTransRDD}\int_{i}^{n_Y} att \neq numeric \tag{6.2}$$

where $\text{Transformation}^{\text{RDD}}$ is the RDD format and funcTransRDD is a function to convert a row y with only categorical attributes from 1 to n on ith index.

While function Eq. 6.3 transforms the categorical value of the attribute to numerical value at the expression evaluation time.

$$\text{Transformation}^{\text{CAT}} = \text{funcTransCat}\int_{i}^{n_Y}(att \text{ in exp}) \tag{6.3}$$

where $\text{Transformation}^{\text{CAT}}$ is the RDD format and funcTransCat is a function to convert a row y with only categorical attributes from 1 to n on ith index and which exist in an expression. These transformations are necessary in order to evaluate expressions from the original ruleset.

6.2.4 UE-RDR Ruleset

Figure 6.3 shows an iris ruleset generated from UE-RDR.

{"defaultclass":"setosa", "model":"UE-RDR-MIN","count":3, "rules":[

{"number":1,"isParent":true,"level":1,"description":"(petal_len > 2.45)","lift":1.5, "cover":100.0, "ok":100.0, "class":"virginica","parentid":0, "childrenNodes":2},

{"number":2,"isParent":false,"level":2,"description":"(petal_len > 2.45) && (petal_len <= 4.95) && (petal_wid <= 1.55)","lift":3.333333, "cover":45.0, "ok":45.0,"class":"versicolor","parentid":1,"isChild":true},

{"number":3,"isParent":false,"level":2,"description":"(petal_len > 2.45) && (petal_wid <= 1.75)", "lift":7.4074, "cover":9.0,"ok":4.0,"class":"versicolor", "parentid":1}]}

Figure 6.3: Iris UE-RDR Ruleset

where “Cover” is the number of instances a rule expression correctly identifies and “Ok” is how many instances (out of the Cover) are correctly classified by this rule. While the Lift is the score for Cover, Ok values and the “count” (total population), determined in Eq. 6.4. While "description" is the rule expression in UEL format.

6.2.5 Lift

In data mining and association rule learning, the Lift (Martinez, 2019) is a measure of the performance of a model (association rule) for prediction or classification as having an enhanced response (with respect to total population), measured against a random choice of the model. So, Lift is the ratio of target response divided by the average response.

For instance, average response rate of a population is 4%, but a segment in a model or rule has a response rate of 12%. Then the Lift score of the segment would be 12% / 4% = 3.0. Let us consider Dataset 1 (Bank dataset) with a distribution of transactions from UK with 4 Fraud and 2 None cases, while 4 Fraud cases from AU. Consider the following rule:

Rule: UK implies Fraud, i.e. IF Country is UK THEN Class = Fraud

$$\text{Lift} = (\text{Ok} / \text{Cover}) / (\text{Cover} / \text{Total}) \quad (6.4)$$

The Lift for the rule using Eq. 6.4 is $(4/6)/(6/10) \approx 1.11$

When Country is UK and Class is Fraud = 4 (OK)

When Country is UK = 6 (COVER)

Total population(instances) = 10 (TOTAL)

while evaluating the expressions of the rules, when multiple rules are true, choosing the predicted class of better Lift score (higher confidence) rule will increase accuracy.

6.2.6 Unified Expressions (UE)

UEL can evaluate mathematical expressions with many operators. It enables dynamic scripting feature. Some of the advantages of UEL is that it supports more than 30 different operators; Rule-based classifiers use only limited operators but using UEL many more operators can be used which are not available in rule-based classifiers, e.g. IN and LIKE Operators. Authors in (Ul Haq et al., 2018) have highlighted the importance of compactness of the prediction model and demonstrated that a compact prediction model is more efficient. The UE will help in ruleset compactness along-with the revised Lift score and hence will improve performance in terms of the time taken for model prediction.

$$\text{Expression}^{UE} = \text{funcUEL}(\text{Expr}^{RDR}) \quad (6.5)$$

where Expression is a UE format and Expr^{RDR} is RDR format expression. funcUEL is a function to convert RDR format expression to UEL format. One of the primary functions of funcUEL is to transform RDR operators and operands to UEL operators and operands.

Few of the transformation are:

Transform “and” to “&&” operator.

Transform “=” to “==” operand.

To make the transformation more generic, profiles are used for transformation operators and operands. Table 6.1 shows the transformation detail.

Table 6.1: RDR and UEL Transformation

RDR	UEL	Category
And	&&	Operator
=	==	Operand

6.2.7 Compactness

The compactness of ruleset can improve the performance of the algorithms and has been proposed in this chapter. One of the challenges was to decide which rules to compact/merge. One of the approaches considered was the nearest neighbour technique using Euclidian based similarity of the instances of two rules. This approach determines (Littin, 1996) distances using Eq. 6.6 and Eq. 6.7:

$$Dp = \sqrt{0.2^2 + 0.3^2} = 0.36 \quad (6.6)$$

$$Dn = \sqrt{0.4^2 + 0.3^2} = 0.5 \quad (6.7)$$

where Dp and Dn are the distances of class p and n respectively. But this technique is computationally expensive, so instead, a customized threshold-based approach is used. The measures and the threshold used in the technique are listed in Table 6.2.

Table 6.2: RDR and UEL Transformation

Measure	Threshold
Nearest Lift score	<= 0.05
Same parent rule	
Smaller expression rule	<=2
IN / BETWEEN operators	> 2

New values of Ok, Cover and Lift score are calculated for merging rules of the customized scheme.

6.2.8 Experimental Setup

A multi-node Hadoop cluster with Spark was set up on a National eResearch Collaboration Tools and Resources (Moloney et al., 2011) research cloud to develop and evaluate this technique for large datasets. Spark is ideal for iterative ML tasks and is much faster than

conventional MapReduce. Figure 6.4 is a typical diagram of Spark internal execution on a Hadoop cluster, which makes it iterative and more efficient than MapReduce.

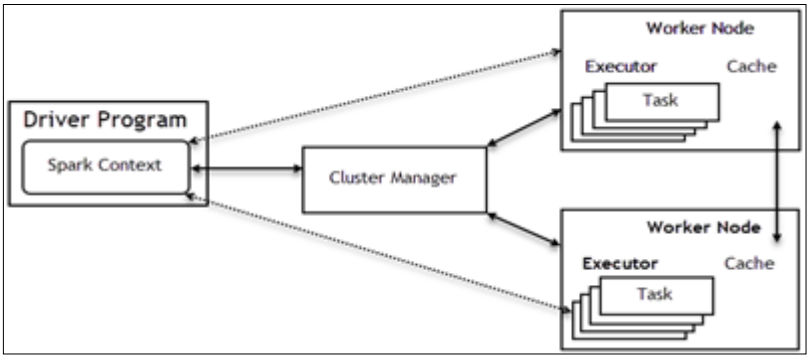


Figure 6.4: Spark Execution Flow

6.2.9 Dataset Characteristics

Characteristics of multiple datasets used for the evaluation are listed in Table 6.3.

Table 6.3: Dataset Characteristics

Dataset	Description	Instances	Features
Dataset 1	Reference Bank Data (Ul Haq et al., 2016)	1,756	14
Dataset 2	Synthetic Bank Data (Ul Haq et al., 2016)	100,000	14
Dataset 3	German Credit Data (Hofmann, 1994; Prasad & Ramakrishna, 2016)	1,000	11
Dataset 4	Credit Approval (Quinlan, 1987, 1992)	691	16
Dataset 5	Adult (Census Income) (Kou et al., 2004; Zadrozny, 2004)	32,562	8

Synthetic Bank data was generated from reference Bank data using HCRUD (Ul Haq et al., 2016) technique. This technique can produce huge dataset on the Hadoop cluster, which is similar to the original reference dataset. The dataset is produced with uniform distribution of class labels, individual and combination of attributes as well. RMSE of the difference of distributions in individual attributes is between 0.00 to 0.78, while the combination of attributes is between 0.80 to 1.85. Spark can use huge datasets, but for evaluation purpose, 100,000 instances of the dataset were used.

6.3 Results

Classification accuracy of UE-RDR technique is compared with existing RDR implementation in Weka (RIDOR). An empirical evaluation was performed with various datasets listed in Table 6.3, with 30% and 70% split for training and testing datasets respectively. Average measurements were taken for various small to large dataset sizes and with five simulation executions. Vertical axes in Figure 6.5 - Figure 6.7 are the percentage of performance improvement of UE-RDR models over the other classifiers. Performance comparison for classification accuracy is shown in Figure 6.5 and Figure 6.7, where the accuracy is the ratio of correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{TP+TN}{\text{TP+FP+FN+TN}} \tag{6.8}$$

where true positives (TP) are the correctly predicted positive values and true negatives (TN) are the correctly predicted negative values, false positives (FP) when actual class is no and predicted class is yes and false negatives (FN) when actual class is yes but predicted class is no.

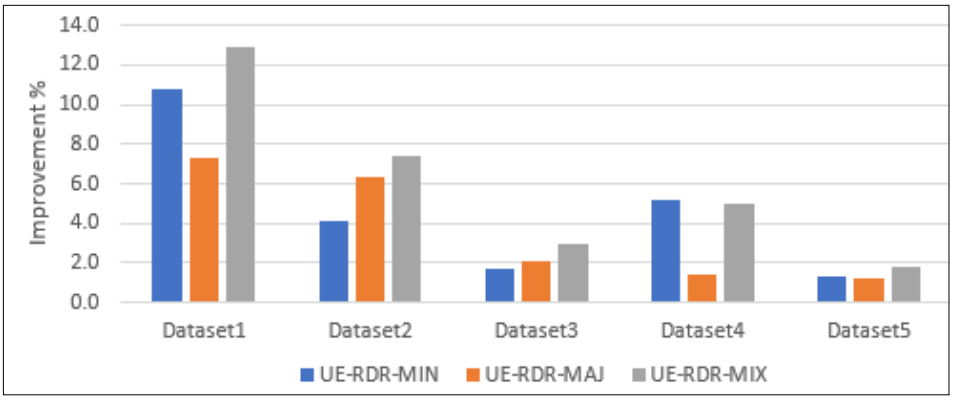


Figure 6.5: % Improvement in Classification Accuracy Over RIDOR

The results show that classification accuracy with all the datasets is improved. Out of the three UE-RDR models, UE-RDR-MIX performance is best among all datasets other than Dataset 4 (Credit Application dataset) where UE-RDR-MIX and UE-RDR-MIN accuracy is almost the same.

Similarly, ruleset compactness results are displayed in Figure 6.6.

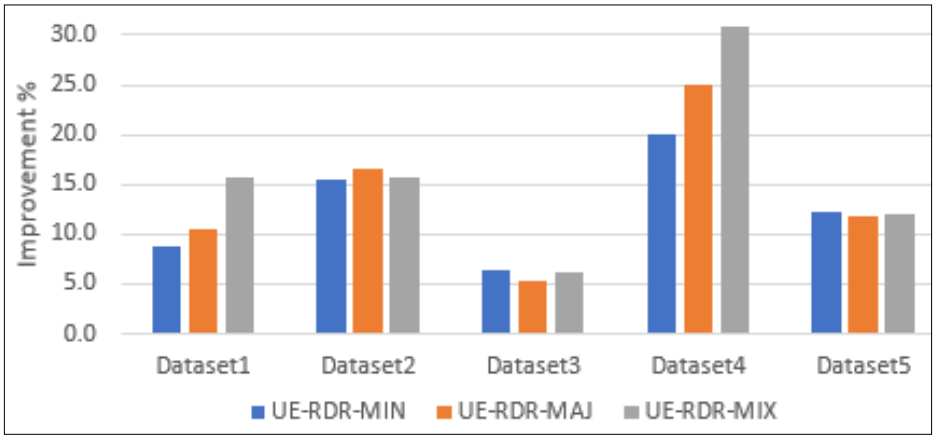


Figure 6.6: % Improvement in Ruleset Compactness Over RIDOR

The results show that compactness with all datasets is improved. However, UE-RDR-MIX compactness is better in Dataset 1 (Bank dataset) and Dataset 2 (Synthetic Bank dataset). For the remaining three datasets, either UE-RDR-MIN or UE-RDR-MAJ models' performance is better.

IPA classifier accuracy for mixed Bank data is compared with UE-RDR-MIX model. Table 6.4 shows that UE-RDR accuracy is higher than IPA classifier.

Table 6.4: Accuracy Comparison with IPA

Technique	**Accuracy**
UE-RDR-MIX	83.76%
IPA(Maruatona, 2013)	73.90%

For further verification, the UE-RDR-MIX classification accuracy is also compared to a non-RDR classifier: Naïve Bayes. Figure 6.7 shows that UE-RDR accuracy is higher than Naïve Bayes accuracy for all datasets, with substantial improvements in accuracy for Datasets 1 and 4.

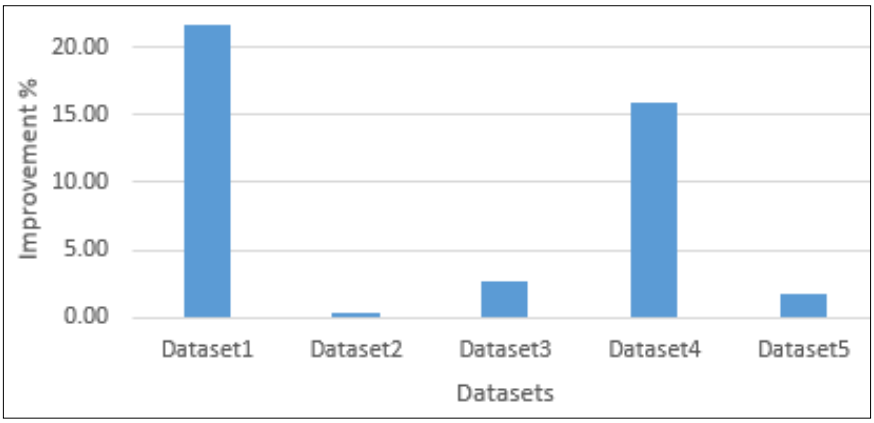


Figure 6.7: % Improvement in Classification Accuracy Over Naïve Bayes

Classification accuracy is compared among the three UE-RDR-models for a specific class label for mixed Bank data. Figure 6.8 shows that classification accuracy is higher with UE-RDR-MIX model.

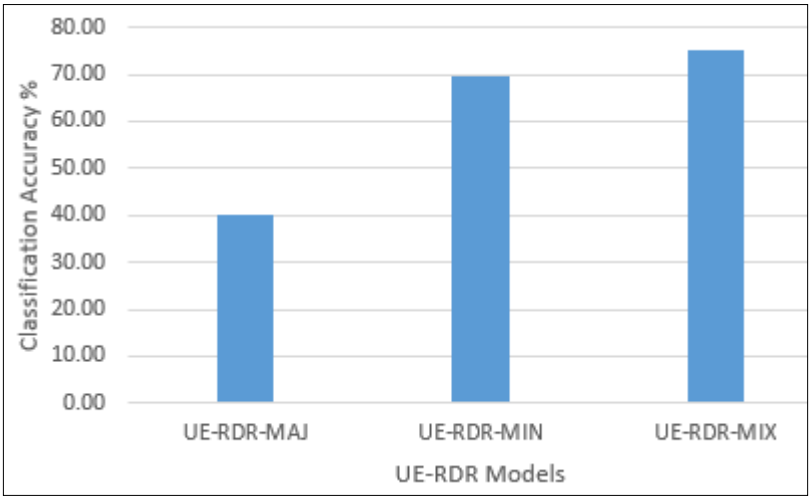


Figure 6.8: Classification Accuracy in Fraud Class Among UE-RDR models

Figure 6.5, Table 6.4 and Figure 6.6 show that UE-RDR-MIX model gives the best classification accuracy. While Figure 6.8 shows that a specific class label which is neither majority class nor minority class, also has a higher classification accuracy with UE-RDR-MIX model. The reason for higher accuracy is because of combined and compact rules in UE-RDR-MIX model for that class from the majority and minority training models.

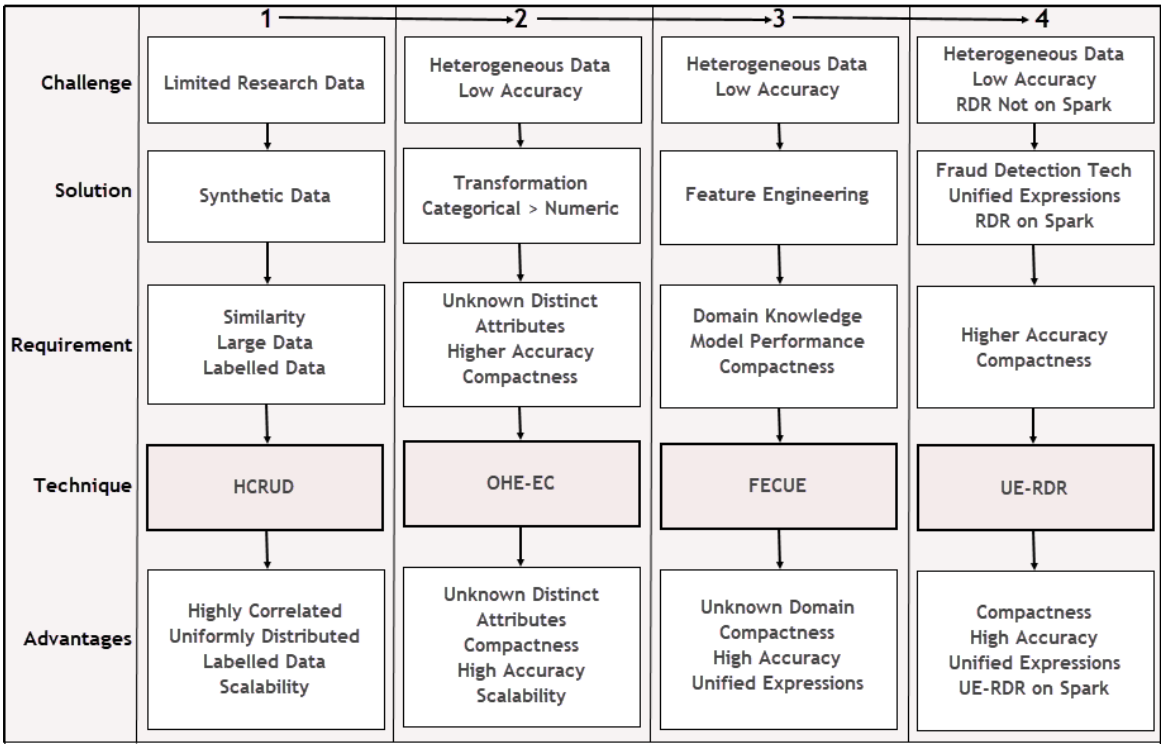
6.4 Conclusion

FD for online banking requires higher classification accuracy for the detection to enhance the confidence of its customers. Out of the available rule-based techniques for FD, RDR is ideal due to its lower maintenance and incremental learning. However, testing and evaluating RDR on distributed and Big data platform is a challenging task, as the RDR classifier has not yet been implemented on Spark. The chapter has shown that the challenge in fraud analysis due to the heterogeneous nature of transactions data (mixed attributes) and Big data can be overcome with UE-RDR. Introducing Unified Expressions in the RDR and evaluating the expressions based on Lift score helped to achieve ruleset compactness and higher accuracy. Further three models, including UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX are also developed in this chapter. UE-RDR-MIX is the most innovative model, which does not exist in RIDOR. It combines and further compacts Majority and Minority class models with least usage of default class and unlike RDR it contains rules of all class labels, so it gives better accuracy from RDR based classifiers.

Classification accuracy is compared with existing RDR implementation: RIDOR. This technique is applied on various datasets including fraud analysis Bank & Synthetic Bank datasets and three publicly available German Credit, Adult (Census Income) and Credit Approval datasets. The empirical evaluation has shown that not only the ruleset size of training and prediction dataset is reduced, but classification accuracy is also improved. Classification accuracy with UE-RDR for Bank dataset is also compared with another RDR based IPA technique and a non-RDR classifier (Naïve Bayes). Results have shown improvement in classification accuracy when compared with these classifiers as well. In this chapter, the developed technique is used for the experimental validation and development of fraud analysis, but it can be used in other domains as well, especially for scalable and distributed systems. Further, this technique can be enhanced for other data formats (LibSVM and ARFF) and a multi-classification system.

Chapter 7

Conclusion and Future Work



Frauds are in various forms and mainly directed at the financial sector, in particular, online banking. (FBI, 2018; Marican & Lim, 2014; Maruatona, 2013; Wei et al., 2013) have indicated the summary of fraud statistics. Taking into account the huge losses to banks and the annual increase in fraud, the identification of online banking fraud has become an important field of study. However, a number of factors are the main obstacles to this research. Limited experimental test data is one of the constraints. Other barriers include public information on fraud analysis, large-scale, distributed and heterogeneous data characteristics. (Bolton & Hand, 2002; Carminati et al., 2015; Maruatona, 2013; Phua et al., 2010) have also highlighted various challenges in fraud analysis research. Limited experimental data and performance improvement on heterogeneous data with various techniques for scalable and distributed data were the main challenges addressed in this research.

The figure above illustrates all the research components linked together. Individual blocks 1, 2, 3 and 4 of that figures relate to Chapter 3, Chapter 4, Chapter 5 and Chapter 6 respectively. The figure shows the research problems, solution, criteria, the technique developed, and the distinctive features of the technique developed.

Large data were needed to carry out a fraud analysis study. However, the lack of availability of massive data and the lack of specific data characteristics was a challenge to start the research. A viable solution to this problem was the generation of synthetic data based on a small labelled sample of actual banking transaction data. Large-scale data, the assignment of appropriate class labels and the similarity of the generated data to the original data were challenges in the synthetic data generation technique. To fill this gap, a Hadoop MapReduce based (ASF, 2015) synthetic data generation technique was developed. The empirical assessment has shown that the produced data has retained a high degree of accuracy and data distribution for single attributes and the combination of attributes and is very similar to the original reference data. The **contributions** in this chapter include:

- Development of a highly correlated rule-based uniformly-distributed synthetic data (HCRUD) technique.

- Evaluate the effectiveness of the method using both the RMSE between the source and synthetic data and in terms of impact on classifier performance.
- The RMSE of the distribution difference for class labels is 0 and in the individual attributes is as close as between 0.00 and 0.78, while the RMSE difference is 0.80 and 1.85 for the combination of attributes.
- The evaluation has shown a high mean classification accuracy of 76.03%, 76.34%, 65.37% and 74.93% with RDR, C45, Naïve Bayes and Random Forest classifiers respectively.
- Performing the evaluation on the multi-node Hadoop cluster.

In the FD field, compact and higher accuracy models are needed as an output from ML algorithms. Generally, the numerical data format provides better results for ML algorithms. However, most banking transactions have categorical or nominal characteristics. In addition, Apache Spark, one of the most renowned large-scale ML systems, recognizes only numerical data. Taking this constraint into account, the transformation of heterogeneous to numerical data was a method of improving performance on heterogeneous data. One-hot-encoding (OHE) (Harris & Harris, 2012) is a commonly used method for converting categorical features to numerical features, but OHE has some challenges, including an increase in data dimensionality, and the fact that the distinct values of the attributes are not always known beforehand. In OHE, each observation indicates the presence (1) or absence (0) of each binary variable. With heterogeneous data limitation in mind, we have developed a technique One-hot Encoded Extended with Compactness (OHE-EC) for categorical features to transform numerical features by compacting sparse-data, although all distinct values are unknown. OHE-EC can be implemented via two models: First Come First Serve (FCFS) and High Distribution First (HDF). The classification accuracy, data size and efficiency in terms of training and predictions models was tested by well-known classifiers including: Random Forest, Decision Tree, Naïve Bayes, SVM and OneVsRest. Alternatively, a synthetic dataset of real bank transactions and the well-known dataset KDD-99 were used for statistical analysis. The **contributions** in this chapter include:

- Developing One-hot Encoded Extended (OHE-E) technique and extending One-hot Encoded Extended with Compactness (OHE-EC).

- Develop two further models: First Come First Serve (FCFS) and High Distribution First (HDF) in One-hot Encoded Extended Compact (OHE-EC).
- Evaluating classification accuracy, the effect on data size and efficiency in terms of the training model and prediction with well-known classification techniques.
- Empirical evaluation with a synthetic dataset generated from real bank transaction data and the well-known KDD-99 dataset.
- After applying OHE-EC on various size bank datasets, classification accuracy improvement with Naïve Bayes, C45 and Random Forest classifiers is between 63% - 65%, 97% - 99% and 97% - 99% respectively. While prediction time improvement with Naïve Bayes, C45, Random Forest, OneVsRest and SVM is upto 69%, 80%, 67%, 22% and 38% respectively.
- Performing the evaluation on the multi-node Hadoop cluster.

FE facilitates the acquisition of additional data by drawing new features from current data. Apart from the categorical conversion of data to numeric data, FE is also one way to improve algorithm performance, but it not only increases data dimensions but also includes comprehensive domain knowledge. The use of FE to detect fraud is an understudied field of research, but our work has shown that it is significant. Feature Engineering and Compact Unified Expressions (FECUE) in an innovative technique presented in this research. FECUE to improve model efficiency through FE with minimal domain knowledge. UE and the use of contextual expressions and the retrieval of geolocation data are another distinct feature of this technique. The use of multiple SPMs has made the technique more generic so that it can be applied to multiple datasets and domains. Empirical evaluation using three well-known classifiers with datasets showed improvement in performance in terms of classification accuracy, precision, recall, f-measuring, time and compactness of the training model. The **contributions** in this chapter include:

- Development of FE technique using custom and configurable SPM when the domain of a dataset is not known in advance.
- Empirical evaluation of the developed technique with multiple datasets.

- After FE with FECUE on various datasets, classification accuracy has improved between 0.93% - 6.75%, 0.32% - 2.64% and 1.29 - 50.58% with RDR, C45 and Random Forest classifiers respectively. While ruleset compactness improvement with RDR and C45 is upto 50% and 15% respectively.
- Ruleset compactness using SPMs.

RBS are commonly used for internet banking FD systems. Online fraud is of different kinds and there are frequent new forms of fraud. The perfect RBS must therefore be able to easily incorporate new patterns of fraud. RDR is an ideal solution for existing rule-based FD systems, due to its lower maintenance and incremental training capability. However, high classification accuracy in mixed datasets and lack of RDR implementation on distributed and Big data platforms are particularly challenging in RDR for scalable data. A Spark-based single classification Unified Expression RDR fraud deduction technique (UE-RDR) for Big data is developed as a solution to these challenges. UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX are the three models designed in the UE-RDR technique. The empirical analysis is performed on a multi-node Hadoop cluster with two RDR based classifiers: RIDOR and IPA and a non-RDR based classifier: Naïve Bayes to validate the proposed models in the technique. Various datasets were used for imperial tests including original Bank data, Synthetic Bank datasets and certain publicly available datasets of the UCI ML repository. The evaluation has shown improvement in classification accuracy and also ruleset compactness. The techniques developed are mainly used to analyze fraud but can be used in other fields, particularly in scalable and distributed systems. The **contributions** in this chapter include:

- Development of a single classification Unified Expression Ripple Down Rules based Fraud Detection (UE-RDR) technique.
- Development of three sub-models: UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX. UE-RDR-MIX is an innovative model for RIDOR, which makes use of majority and minority classes and multi-level compactness.
- Evaluation of the developed technique for classification accuracy and ruleset compactness with multiple datasets and comparison with various RDR and non-RDR based classifiers.

- Improvement in classification accuracy when compared with Naïve Bayes, RIDOR and IPA is upto 22%, 30% and 13% respectively.
- Implementation of the developed technique on distributed and Big data ML platform, Spark.
- Performing the evaluation on multi-node Hadoop cluster, demonstrating the applicability of the approach to distributed systems.

7.1 Limitations

The research was conducted with certain limitations and constraints.

- The work is specifically used for online banking FD, but it can be used to investigate or to detect fraud in any area.
- In synthetic data generation technique (HCRUD), a standard method was employed. The same approach can be used to produce synthetic data from any dataset; however for an analysis purpose, it was implemented with reference data given by a partner bank.
- A multi-node Cloudera Hadoop cluster was configured on a research cloud with limited available resource nodes. A larger Hadoop cluster with a larger number of worker and data nodes and a higher specification name node and resource manager can also be used to process much larger datasets. Hadoop cluster was used for the development and the evaluation of HCRUD, OHE-EC and UE-RDR techniques.

7.2 Future Research

There is still room to extend this research. Synthetic data generation technique (HCRUD) can be extended to generate data with descriptive language where only the attribute and class distributions are defined. UE-RDR technique can be enhanced for other data formats (LibSVM and ARFF). Since online banking FD is a single class domain, so for the time being a single classification classifier was developed. This technique can also be enhanced for the multi-classification system. More testing could be done to evaluate the behaviour of the developed techniques on high-dimensional data or the datasets having higher distinct values of categorical attributes. The research can also be extended for real-time streaming

data where synthetic data generation technique can serve as a virtual bank to produce real-time streams. One of the proposed future research is to implement prudence on cloud-based systems having multiple administrators and re-training the training model based on the feedback received from the administrator for un-handled cases. For evaluation purpose, a multi-node, a Cloudera Hadoop cluster was configured and used in this research, but the techniques can be evaluated on other platforms and flavours of Hadoop including Apache Hadoop, Hortonworks, MapR and AWS EMR.

References

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications, 68*, 90-113.

ACI. (2011). Proactive Risk Manager. Retrieved from https://www.aciworldwide.com/products/proactive-risk-manager

AFP. (2015). Online fraud and scams. Retrieved from http://www.afp.gov.au/policing/cybercrime/online-fraud-and-scams

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning, 6*(1), 37-66.

Altexsoft. (2017). Fraud Detection: How Machine Learning Systems Help Reveal Scams in Fintech, Healthcare, and eCommerce. Retrieved from https://www.altexsoft.com/whitepapers/fraud-detection-how-machine-learning-systems-help-reveal-scams-in-fintech-healthcare-and-ecommerce/

Anderberg, M. R. (1973). *Cluster Analysis for Applications*. New York: Academic Press.

APF. (2018). Western Australia Police Force. Retrieved from https://www.police.wa.gov.au/

ASF. (2004). Java Expression Language (JEXL). Retrieved from http://commons.apache.org/proper/commons-jexl

ASF. (2012). Apache Spark. Retrieved from https://spark.apache.org/

ASF. (2015). Apache Hadoop. Retrieved from http://hadoop.apache.org/

Ayala-Rivera, V., McDonagh, P., Cerqueus, T., & Murphy, L. (2013). Synthetic Data Generation using Benerator Tool. *arXiv preprint arXiv:1311.3312*, 12.

Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications, 51*(C), 134-142. doi:10.1016/j.eswa.2015.12.030

Bai, J. (2013). *Feasibility analysis of big log data real time search based on Hbase and ElasticSearch.* Paper presented at the 2013 Ninth International Conference on Natural Computation (ICNC), Shenyang, China.

Bakar, Z. A., Mohemad, R., Ahmad, A., & Deris, M. M. (2006). *A comparative study for outlier detection techniques in data mining.* Paper presented at the 2006 IEEE conference on cybernetics and intelligent systems.

Behdad, M., Barone, L., Bennamoun, M., & French, T. (2012). Nature-inspired techniques in the context of fraud detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(6), 1273-1290.

Ben-Gal, I. (2005). Outlier detection *Data mining and knowledge discovery handbook* (pp. 131-146): Springer.

Bergmann, V. (2015). Databene Benerator. Retrieved from http://databene.org/databene-benerator

Bijak, K., & Thomas, L. C. (2012). Does segmentation always improve model performance in credit scoring? *Expert Systems with Applications, 39*(3), 2433-2442.

Bolton, R. J., & Hand, D. J. (2002). Statistical Fraud Detection: A Review. *Statistical Science, 17*(3), 235-255.

Boriah, S., Chandola, V., & Kumar, V. (2008). *Similarity measures for categorical data: A comparative evaluation.* Paper presented at the Proceedings of the 2008 SIAM international conference on data mining. Society for Industrial and Applied Mathematics.

Brabazon, A., Cahill, J., Keenan, P., & Walsh, D. (2010). *Identifying online credit card fraud using Artificial Immune Systems*.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32. doi:10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification And Regression Trees*: Routledge.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). *LOF: identifying density-based local outliers.* Paper presented at the ACM sigmod record.

Buczak, A. L., Babin, S., & Moniz, L. (2010). Data-driven approach for creating synthetic electronic medical records. *BMC Medical Informatics and Decision Making*, *10*(1), 59-59. doi:10.1186/1472-6947-10-59

Capterra. (2019). Fraud Management Software. Retrieved from https://www.capterra.com.au/directory/10058/financial-fraud-detection/software

Carminati, M., Caron, R., Maggi, F., Epifani, I., & Zanero, S. (2015). BankSealer: A decision support system for online banking fraud analysis and investigation. *Computers & Security, 53*(C), 175-186. doi:10.1016/j.cose.2015.04.002

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES, 1*(4), 300-307.

Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*(2), 50.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR), 41*(3), 1-58. doi:10.1145/1541880.1541882

Chang, C.-c., & Lin, C.-j. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 1-27. doi:10.1145/1961189.1961199

Chauhan, N. K., & Singh, K. (2018). *A Review on Conventional Machine Learning vs Deep Learning.* Paper presented at the International Conference on Computing, Power and Communication Technologies (GUCON).

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of artificial intelligence research, 16*, 321-357.

Chen, C.-M., Guan, D. J., Huang, Y.-Z., & Ou, Y.-H. (2012). *Attack Sequence Detection in Cloud Using Hidden Markov Model.* Paper presented at the 2012 Seventh Asia Joint Conference on Information Security, Tokyo.

Chen, W. (2016). *Learning with Scalability and Compactness.* (PHD PHD), Washington University in St. Louis.

Chilo, J., Horvath, G., Lindblad, T., & Olsson, R. (2009). *Electronic Nose Ovarian Carcinoma Diagnosis Based on Machine Learning Algorithms.* Paper presented at the Industrial Conference on Data Mining, Berlin, Heidelberg.

Chiu, C.-Y., Yeh, C.-T., & Lee, Y.-J. (2013). *Frequent Pattern Based User Behavior Anomaly Detection for Cloud System.* Paper presented at the 2013 Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taipei, Taiwan.

Christen, P., & Pudjijono, A. (2009). *Accurate synthetic generation of realistic personal information.* Paper presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining.

Christen, P., & Vatsalan, D. (2013). *Flexible and extensible generation and corruption of personal data.* Paper presented at the Proceedings of the 22nd ACM international conference on Information & Knowledge Management.

Compton, P. (2011). *Pacific Knowledge Systems: Challenges with Rules*. Retrieved from Sydney: http://pks.com.au/wp-content/uploads/2015/03/WhitePaperChallengesWithRulesPKS.pdf

Compton, P., & Jansen, R. (1988). *Knowledge in context: A strategy for expert system maintenance.* Paper presented at the Australian Joint Conference on Artificial Intelligence, Adelaide, Australia.

Compton, P., Preston, P., Edwards, G., & Kang, B. (1996). *Knowledge based systems that have some idea of their limits.* Paper presented at the Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop, Sydney.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.

Coyle, E. J., Roberts, R. G., Collins Jr., E. G., & Barbu, A. (2013). Synthetic data generation for classification via uni-modal cluster interpolation. *Autonomous Robots, 37*(1), 27-45. doi:10.1007/s10514-013-9373-9

Dazeley, R., & Kang, B.-H. (2008). *Detecting the knowledge boundary with prudence analysis.* Paper presented at the Australasian Joint Conference on Artificial Intelligence, Berlin, Heidelberg.

Dazeley, R., Warner, P., Johnson, S., & Vamplew, P. (2010). *The ballarat incremental knowledge engine.* Paper presented at the 11th International Workshop on Knowledge Management and Acquisition for Smart Systems and Services, Berlin, Heidelberg.

Dazeley, R. P. (2006). *To The Knowledge Frontier and Beyond: A Hybrid System for Incremental Contextual-Learning and Prudence Analysis.* (PHD PHD), University of Tasmania.

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM, 51*(1), 107-113. doi:10.1145/1327452.1327492

Dehaven, V. R. (2014). Machine Learning: Future Capabilities and their Implications.

Demilli, R. A., & Offutt, A. J. (1991). Constraint-based automatic test data generation. *IEEE Transactions on Software Engineering, 17*(9), 900-910. doi:10.1109/32.92910

Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., & Tan, P.-N. (2002). *Data mining for network intrusion detection.* Paper presented at the Proc. NSF Workshop on Next Generation Data Mining.

Duman, E., & Ozcelik, M. H. (2011). Detecting credit card fraud by genetic algorithm and scatter search. *Expert Systems with Applications, 38*(10), 13057-13063. doi:https://doi.org/10.1016/j.eswa.2011.04.110

Durrant, B., Frank, E., Hunt, L., Holmes, G., Mayo, M., Pfahringer, B., . . . Witten, I. (2018). An ARFF (Attribute-Relation File Format). Retrieved from https://waikato.github.io/weka-wiki/arff_stable/

El Bouchti, A., Chakroun, A., Abbar, H., & Okar, C. (2017). *Fraud detection in banking using deep reinforcement learning.* Paper presented at the Seventh International Conference on Innovative Computing Technology (INTECH).

FBI. (2018). Internet Crime Complaint Center, 2018*2019*(20/08/2019). Retrieved from https://pdf.ic3.gov/2018_IC3Report.pdf

FICO. (2010). Fico Application Fraud Manager. Retrieved from https://www.fico.com/en/products/fico-application-fraud-manager

FraudNet. (1997). Enterprise Fraud Prevention. Retrieved from https://fraud.net

Fürnkranz, J., & Widmer, G. (1994). Incremental reduced error pruning *Machine Learning Proceedings 1994* (pp. 70-77): Elsevier.

Gaines, B. R., & Compton, P. (1995). Induction of ripple-down rules applied to modeling large databases. *Journal of Intelligent Information Systems, 5*(3), 211-228.

Garla, V. N., & Brandt, C. (2012). Ontology-guided feature engineering for clinical text classification. *Journal of biomedical informatics, 45*(5), 992-998.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*: MIT press.

Hackeling, G. (2014). *Mastering Machine Learning with scikit-learn* (A. S. Roshni Banerjee Ed.). Birmingham B3 2PB, UK: Packt Publishing Ltd.

Hafiz, K. T., Aghili, S., & Zavarsky, P. (2016). *The use of predictive analytics technology to detect credit card fraud in Canada*.

Haigh, J. a. (2013). *Probability Models* (2nd ed.): Springer.

Harris, D. M., & Harris, S. L. (2012). *Digital Design and Computer Architecture* (N. McFadden Ed. 2nd ed.). USA: Morgan Kaufmann.

Hartigan, J. A. (1975). *Cluster algorithms* (Vol. 214).

Harutyunyan, A. N., Poghosyan, A. V., Grigoryan, N. M., & Marvasti, M. A. (2014). *Abnormality analysis of streamed log data.* Paper presented at the 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow.

Hawkins, D. M. (1980). *Identification of outliers* (Vol. 11): Springer.

Hayes, M. A., & Capretz, M. A. M. (2014). *Contextual Anomaly Detection in Big Sensor Data.* Paper presented at the 2014 IEEE International Congress on Big Data, Anchorage, AK.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep Learning in Finance. *arXiv preprint arXiv:1602.06561*.

Herland, M., Khoshgoftaar, T., & Bauder, R. (2018). Big Data fraud detection using multiple medicare data sources. *Journal of Big Data, 5*(1), 1-21. doi:10.1186/s40537-018-0138-3

Herschel, G., Linden, A., & Kart, L. (2015). Magic quadrant for advanced analytics platforms. *Gartner Report G, 270612*.

Hodge, V., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review, 22*(2), 85-126. doi:10.1023/B:AIRE.0000045502.10941.a9

Hofmann, H. (1994). *Statlog (German Credit Data) Data Set* [Multivariate]. Financial. Retrieved from: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2010). A Practical Guide to Support Vector Classification. 16.

Huang, A. (2008). *Similarity measures for text document clustering.* Paper presented at the Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand.

Huang, Z. (1997). *Clustering large data sets with mixed numeric and categorical values.* Paper presented at the Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining(PAKDD).

Huang, Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery, 2*, 283–304.

Ilgun, K., Kemmerer, R. A., & Porras, P. A. (1995). State transition analysis: a rule-based intrusion detection approach. *IEEE Transactions on Software Engineering, 21*(3), 181-199. doi:10.1109/32.372146

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. NJ: Prentice-Hall.

Jeske, D. R., Samadi, B., Lin, P. J., Ye, L., Cox, S., Xiao, R., . . . Rich, R. (2005). *Generation of synthetic data sets for evaluating the accuracy of knowledge discovery systems.* Paper presented at the Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.

Jian, S., Cao, L., Pang, G., Lu, K., & Gao, H. (2017). *Embedding-based representation of categorical data by hierarchical value coupling learning.* Paper presented at the Proceedings of the 26th International Joint Conference on Artificial Intelligence.

Jin, H., Chen, J., He, H., Kelman, C., McAullay, D., & O'Keefe, C. M. (2010). Signaling potential adverse drug reactions from administrative health databases. *IEEE Transactions on knowledge and data engineering, 22*(6), 839-853.

John, S. N., Kennedy, O. O., Kennedy, C. G., Anele, C., & Olajide, F. (2016). *Realtime Fraud Detection in the Banking Sector Using Data Mining Techniques/Algorithm.* Paper presented at the 2016 International Conference on Computational Science and Computational Intelligence, Las Vegas, Nevada, USA.

Joshi, M. V., Karypis, G., & Kumar, V. (1998). *ScalParC : A New Scalable and Efficient Parallel Classification Algorithm for Mining Large Datasets.* Paper presented at the Parallel processing symposium and symposium on parallel and distributed processing, IPPS/SPDP 1998., Orlando.

Kambatla, K., Kollias, G., Kumar, V., & Grama, A. (2014). Trends in big data analytics. *Journal of Parallel and Distributed Computing, 74*(7), 2561-2573. doi:10.1016/j.jpdc.2014.01.003

Kang, B. H., Compton, P., & Preston, P. (1995). *Multiple Classification Ripple Down Rules: Evaluation and Possibilities.* Paper presented at the 9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, Banff.

Kao, W.-C., Chung, K.-M., Sun, C.-L., & Lin, C.-J. (2004). Decomposition methods for linear support vector machines. *Neural Computation, 16*(8), 1689-1704.

Kazemi, Z., & Zarrabi, H. (2017). *Using deep networks for fraud detection in the credit card transactions.* Paper presented at the 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI).

Keen, B. (2015). Generate Data. Retrieved from http://www.generatedata.com/

Kelarev, A., Dazeley, R., Stranieri, A., Yearwood, J., & Jelinek, H. (2012). *Detection of CAN by ensemble classifiers based on ripple down rules.* Paper presented at the Pacific Rim Knowledge Acquisition Workshop.

Khan, S., Shakil, K. A., & Alam, M. (2015). Cloud-based big data analytics—a survey of current research and future directions. *arXiv preprint arXiv:1508.04733,* 595-604.

Khorshed, M. T., Shawkat Ali, A. B. M., & Wasimi, S. A. (2012). A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud

computing. *Future Generation Computer Systems, 28*(6), 833-851. doi:10.1016/j.future.2012.01.006

Kim, Y. S., Compton, P., & Kang, B. H. (2012). *Ripple-down rules with censored production rules.* Paper presented at the Pacific Rim Knowledge Acquisition Workshop.

Kou, Y., Lu, C.-T., Sirwongwattana, S., & Huang, Y.-P. (2004). *Survey of fraud detection techniques.* Paper presented at the IEEE International Conference on Networking, Sensing and Control, 2004.

Kount. (2006). Fraud Prevention Solution. Retrieved from https://www.kount.com

Kovach, S., & Ruggiero, W. V. (2011). *Online banking fraud detection based on local and global behavior.* Paper presented at the Proc. of the Fifth International Conference on Digital Society, Guadeloupe, France.

Lavion, D. (2018). Global Economic Crime and Fraud Survey 2018, 30. Retrieved from Pulling fraud out of the shadows website: https://www.pwc.com/gx/en/forensics/global-economic-crime-and-fraud-survey-2018.pdf

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature, 521*(7553), 436-444.

Lee, J.-H., Park, M.-W., Eom, J.-H., & Chung, T.-M. (2011). *Multi-level Intrusion Detection System and log management in Cloud Computing.* Paper presented at the 13th International Conference on Advanced Communication Technology (ICACT2011), Seoul.

Lenat, D. (2006). Creativity vs. Common Sense. In L. Weiss (Ed.). CA, USA: USC ICT.

Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., & Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36*(1), 16-24.

Lin, P. J., Samadi, B., & Jeske, D. R. (2006, 10/04/2006). *Development of a Synthetic Data Set Generator for Building and Testing Information Discovery Systems.* Paper presented at the Third International Conference on Information Technology: New Generations (ITNG'06).

LIN, X., WANG, P., & WU, B. (2013). *Log analysis in cloud computing environment with Hadoop and Spark.* Paper presented at the 2013 5th IEEE International Conference on Broadband Network & Multimedia Technology, Guilin.

Littin, J. N. (1996). *Learning relational ripple-down rules.* (PHD PHD), University of Waikato, Hamilton New Zealand. Retrieved from http://www.cs.waikato.ac.nz/~ml/publications/1996/JLittin96-Thesis.pdf

Maj, P. (2003). DBMonster Core. Retrieved from http://dbmonster.sourceforge.net/

Marican, L., & Lim, S. (2014). Microsoft Consumer Safety Index reveals impact of poor online safety behaviours in Singapore. Retrieved from https://news.microsoft.com/en-sg/2014/02/11/microsoft-consumer-safety-index-reveals-impact-of-poor-online-safety-behaviours-in-singapore/

Martinez, G. (2019). Lift (data mining). Retrieved from https://en.wikipedia.org/wiki/Lift_(data_mining)

Maruatona, O. (2013). *Internet banking fraud detection using prudent analysis.* (PHD PHD), University of Ballarat.

Maruatona, O., Vamplew, P., & Dazeley, R. (2012). *RM and RDM, a Preliminary Evaluation of Two Prudent RDR Techniques.* Paper presented at the Pacific Rim Knowledge Acquisition Workshop.

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

Melo-Acosta, G. E., Duitama-Munoz, F., & Arias-Londono, J. D. (2017). *Fraud detection in big data using supervised and semi-supervised learning techniques.* Paper presented at the 2017 IEEE Colombian Conference on Communications and Computing (COLCOM, Cartagena, Colombia.

Meng, X. (2014). Sparse data support in MLlib, 1-40. Retrieved from https://spark-summit.org/2014/

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., . . . Talwalkar, A. (2016). Mllib: Machine learning in apache spark. *Journal of Machine Learning Research, 17*(34), 1-7.

Moloney, G., Barker, M., Coddington, P., & Mecoles, K. (2011). NECTAR. Retrieved from https://nectar.org.au

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559-569.

Oxford. (Ed.) (2011) Concise Oxford English Dictionary (12th ed.). United Kingdom: Oxford University Press.

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

Patel, A., Taghavi, M., Bakhtiyari, K., & Celestino Júnior, J. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications*, *36*(1), 25-41.

PatternSpy. (2015). PatternSpy For Banking - Fraud Management Software. Retrieved from https://www.patternspy.tech/

Pentreath, N. (2015). *Machine learning with spark*: Packt Publishing Ltd.

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

Prasad, Y. A. S., & Ramakrishna, D. G. (2016). A novel probabilistic based feature selection model for credit card anomaly detection. *Journal of Theoretical and Applied Information Technology*, *94*(2), 335.

Prayote, A. (2007). *Knowledge Based Anomaly Detection.* (PHD PHD), University of NSW.

PwC. (2016). Global Economic Crime Survey 2016, 56. Retrieved from Adjusting the Lens on Economic Crime website: https://www.pwc.com/gx/en/economic-crime-survey/pdf/GlobalEconomicCrimeSurvey2016.pdf

Qian, Y., Li, F., Liang, J., Liu, B., & Dang, C. (2016). Space structure and clustering of categorical data. *IEEE transactions on neural networks and learning systems*, *27*(10), 2047-2059. doi:10.1109/TNNLS.2015.2451151

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning, 1*, 81–106. doi:https://doi.org/10.1007/BF00116251

Quinlan, J. R. (1987). Simplifying decision trees. *International journal of man-machine studies*, *27*(3), 221-234.

Quinlan, J. R. (1992). *C4.5: Programs for Machine Learning* (Ist ed.): Morgan Kaufmann.

Ré, C., Sadeghian, A. A., Shan, Z., Shin, J., Wang, F., Wu, S., & Zhang, C. (2014). Feature Engineering for Knowledge Base Construction. *arXiv preprint arXiv:1407.6439*.

Richards, D. (2003). Knowledge-based system explanation: The ripple-down rules alternative. *Knowledge and Information Systems, 5*(1), 2-25. doi:10.1007/s10115-002-0076-3

Richards, D. (2009). Two decades of Ripple Down Rules research. *The Knowledge Engineering Review, 24*(2), 159-184. doi:10.1017/S0269888909000241

RiskNet. (1998). Fraud Managed Services. Retrieved from https://www.aicorporation.com/products-services/fraud-managed-services/

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. (2018). *Deep learning detecting fraud in credit card transactions.* Paper presented at the Systems and Information Engineering Design Symposium (SIEDS).

Rubin, D. B. (1993). Statistical disclosure limitation. *Journal of official Statistics, 9*(2), 461-468.

Ryza, S., Laserson, U., Owen, S., & Wills, J. (2015). *Advanced Analytics with Spark* (M. Beaugureau Ed. Ist ed.): O'Reilly Media, Inc.

Sánchez-Marono, N., Alonso-Betanzos, A., García-González, P., & Bolón-Canedo, V. (2010). *Multiclass classifiers vs multiple binary classifiers using filters for feature selection.* Paper presented at the The 2010 International Joint Conference on Neural Networks (IJCNN).

Sarawat, A., Yang, S. K., Byeong, H. K., & Qing, L. (2015). *Schema Mapping Using Hybrid Ripple - Down Rules.* Paper presented at the Proceedings of the 38th Australasian Computer Science Conference(ACSC2015), Sydney, Australia.

SAS. (2007). SAS Fraud Management. Retrieved from https://www.sas.com/en_au/software/fraud-management.html

Shanahan, J. G., & Dai, L. (2015). *Large scale distributed data science using apache spark.* Paper presented at the Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia.

Sharma, A., & Panigrahi, P. K. (2013). A review of financial accounting fraud detection based on data mining techniques. *International journal of computer applications, 39*(1), 37-47.

Shih, M.-Y., Jheng, J.-W., & Lai, L.-F. (2010). A two-step method for clustering mixed categroical and numeric data. *Tamkang Journal of Science and Engineering, 13*(1), 11-19.

Swain, S. R., & Sarangi, S. S. (2013). Study of Various Classification Algorithms using Data Mining. *International Journal of Advanced Research in Science and Technology (IJARST), 2*(2), 110-114.

Syed, N. A., Huan, S., Kah, L., & Sung, K. (1999). Incremental learning with support vector machines. *Citeseer*.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). *A detailed analysis of the KDD CUP 99 data set.* Paper presented at the Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. , Ottawa, Canada.

Tijms, H. C. (2012). *Understanding probability* (3rd Ed.). New York: Cambridge University Press.

Turner, C. R., Fuggetta, A., Lavazza, L., & Wolf, A. L. (1999). A conceptual basis for feature engineering. *The Journal of Systems & Software, 49*(1), 3-15. doi:10.1016/S0164-1212(99)00062-X

Ul Haq, I., Gondal, I., & Vamplew, P. (2019). *Enhancing Model Performance for Fraud Detection by Feature Engineering and Compact Unified Expressions.* Paper presented at the 19th International Conference on Algorithms and Architectures for Parallel Processing, Melbourne, Australia.

Ul Haq, I., Gondal, I., & Vamplew, P. (2020). *Unified Expression Ripple Down Rules based Fraud Detection Technique for Scalable Data.* Paper presented at the AUSTRALASIAN INFORMATION SECURITY CONFERENCE (AISC 2020), Melbourne, Victoria, Australia.

Ul Haq, I., Gondal, I., Vamplew, P., & Brown, S. (2018). *Categorical Features Transformation with Compact One-hot Encoder for Fraud Detection in Distributed Environment.* Paper presented at the The 16th Australasian Data Mining Conference, Bathurst NSW, Australia.

Ul Haq, I., Gondal, I., Vamplew, P., & Layton, R. (2016). *Generating Synthetic Datasets for Experimental Validation of Fraud Detection.* Paper presented at the Fourteenth Australasian Data Mining Conference, Canberra, Australia.

Vastenburg, M. H. (2004). *SitMod: A Tool for Modeling and Communicating Situations*.

Vernekar, S. S., & Buchade, A. (2013). *MapReduce based log file analysis for system threats and problem identification.* Paper presented at the 2013 3rd IEEE International Advance Computing Conference, Pune.

Waikato. (1993). Data Mining Software in Java. Retrieved from http://www.cs.waikato.ac.nz/ml/weka/

Wei, W., Li, J., Cao, L., Ou, Y., & Chen, J. (2013). Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web, 16*(4), 449-475. doi:10.1007/s11280-012-0178-0

White, T. (2015). *Hadoop: The Definitive Guide* (M. Loukides & M. Blanchette Eds. 4th ed.): Oreily.

Wikipedia. (2017). One-hot encoding. Retrieved from https://en.wikipedia.org/wiki/One-hot

Wisser, R. (2015). Open Jail - The Jailer Project. Retrieved from http://jailer.sourceforge.net/

Witten, I. H., Frank, E., Hall, M., & Pal, C. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.): Elsevier Science & Technology.

Xiaofeng, Z., Shichao, Z., Zhi, J., Zili, Z., & Zhuoming, X. (2011). Missing Value Estimation for Mixed-Attribute Data Sets. *Knowledge and Data Engineering, IEEE Transactions on, 23*(1), 110-121. doi:10.1109/TKDE.2010.99

Xu, Y., Hong, K., Tsujii, J., & Chang, E. I.-C. (2012). Feature engineering combined with machine learning and rule-based methods for structured information extraction from

narrative clinical discharge summaries. *Journal of the American Medical Informatics Association, 19*(5), 824-832. doi:10.1136/amiajnl-2011-000776

Yang, S. K., Sung, S. P., Edward, D., & Byeong, H. K. (2004). *Adaptive Web Document Classification with MCRDR.* Paper presented at the Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04).

Yoo, S., & Harman, M. (2012). Test data regeneration: generating new test data from existing test data. *Software Testing, Verification and Reliability, 22*(3), 171-201. doi:10.1002/stvr.435

Yu, H.-F., Lo, H.-Y., Hsieh, H.-P., Lou, J.-K., McKenzie, T. G., Chou, J.-W., . . . Wei, Y.-H. (2010). *Feature engineering and classifier ensemble for KDD cup 2010.* Paper presented at the KDD Cup 2010.

Zadrozny, B. (2004). *Learning and evaluating classifiers under sample selection bias.* Paper presented at the Proceedings of the twenty-first international conference on Machine learning.

Zhang, K., & Jin, H. (2010). *An effective pattern based outlier detection approach for mixed attribute data.* Paper presented at the Australasian Joint Conference on Artificial Intelligence.

Zhang, L., Xiong, X., Zhao, S., Botelho, A., & Heffernan, N. T. (2017). *Incorporating rich features into deep knowledge tracing.* Paper presented at the Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale.

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion, 42*, 146-157. doi:10.1016/j.inffus.2017.10.006

Zhang, Y., Meratnia, N., & Havinga, P. (2010). Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials, 12*(2), 159-170.

Zhou, X., & Xiang, W. (2015). Predict Employees' Computer Access Needs in Company.

List of Appendices

Appendix A: Dataset Samples

KDD-99 Dataset

Protocol type	service	flag	Src bytes	Dst bytes	Lroot shell	Serror rate	Srv Serror rate	Rerror rate	Srv Rerror Rate	Class
tcp	http	SF	54540	8314	0	0	0	0	0.5	back
tcp	http	SF	54540	8314	0	0	0	0	0.33	back
tcp	http	SF	54540	8314	0	0	0	0	0.25	back
tcp	http	SF	54540	8314	0	0	0	0	0	back
tcp	http	SF	54540	8314	0	0	0	0	0	back
tcp	http	SF	54540	8314	0	0	0	0	0.2	back
tcp	telnet	RSTO	0	15	0	0	0	1	1	ipsweep
tcp	private	REJ	0	0	0	0	0	1	1	ipsweep
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	finger	S0	0	0	0	1	1	0	0	land
tcp	private	S0	0	0	0	1	1	0	0	neptune
tcp	private	S0	0	0	0	1	1	0	0	neptune
tcp	private	S0	0	0	0	1	1	0	0	neptune

German Credit Dataset

Over Draft	Credit Usage	Current Balance	Location	Other Parties	Cc Age	Other Payment Plans	Housing	Own Telephone	Class
<0	6	1169	4	none	67	none	Own	yes	good
<0	24	4870	3	none	53	none	'for free'	none	bad
no checking	36	9055	2	none	35	none	'for free'	yes	good
no checking	9	2134	4	none	48	none	own	yes	good
<0	6	2647	2	none	44	none	rent	none	good
<0	10	2241	1	none	48	none	rent	none	good
no checking	6	426	4	none	39	none	own	none	good
>=200	12	409	3	none	42	none	rent	none	good
0<=X<200	7	2415	3	guarantor	34	none	own	none	good
<0	60	6836	3	none	63	none	own	yes	bad
0<=X<200	18	1913	3	none	36	bank	own	yes	good
<0	24	4020	2	none	27	stores	own	none	good
0<=X<200	18	5866	2	none	30	none	own	yes	good
>=200	12	1474	4	none	33	bank	own	yes	good
0<=X<200	45	4746	4	none	25	none	own	none	bad
no checking	48	6110	1	none	31	bank	'for free'	yes	good
>=200	18	2100	4	'co applicant'	37	stores	own	none	bad
>=200	10	1225	2	none	37	none	own	yes	good
0<=X<200	9	458	4	none	24	none	own	none	good
no checking	30	2333	4	none	30	bank	own	none	good

Adult Census Income Dataset

Age	Work Class	Fnl Wgt	Education	Education Num	Sex	Capital Gain	Capital Loss	Hours Per Week	Class
39	State-gov	77516	Bachelors	13	Male	2174	0	40	LTE50K
53	Private	234721	11th	7	Male	0	0	40	LTE50K
37	Private	284582	Masters	14	Female	0	0	40	LTE50K
34	Private	245487	7th-8th	4	Male	0	0	45	LTE50K
25	Self-emp-not-inc	176756	HS-grad	9	Male	0	0	35	LTE50K
32	Private	186824	HS-grad	9	Male	0	0	40	LTE50K
38	Private	28887	11th	7	Male	0	0	50	LTE50K
40	Private	193524	Doctorate	16	Male	0	0	60	GT50K
54	Private	302146	HS-grad	9	Female	0	0	20	LTE50K
35	Federal-gov	76845	9th	5	Male	0	0	40	LTE50K
43	Private	117037	11th	7	Male	0	2042	40	LTE50K
59	Private	109015	HS-grad	9	Female	0	0	40	LTE50K
56	Local-gov	216851	Bachelors	13	Male	0	0	40	GT50K
19	Private	168294	HS-grad	9	Male	0	0	40	LTE50K
39	Private	367260	HS-grad	9	Male	0	0	80	LTE50K
49	Private	193366	HS-grad	9	Male	0	0	40	LTE50K
23	Local-gov	190709	Assoc-acdm	12	Male	0	0	52	LTE50K
20	Private	266015	Some-college	10	Male	0	0	44	LTE50K

Credit Approval Dataset

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	Class
b	30.83	0	u	g	w	v	1.25	t	t	1	f	g	202	0	YES
a	58.67	4.46	u	g	q	h	3.04	t	t	6	f	g	43	560	YES
a	24.5	0.5	u	g	q	h	1.5	t	f	0	f	g	280	824	YES
b	27.83	1.54	u	g	w	v	3.75	t	t	5	t	g	100	3	YES
b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	120	0	YES
b	32.08	4	u	g	m	v	2.5	t	f	0	t	g	360	0	YES
b	33.17	1.04	u	g	r	h	6.5	t	f	0	t	g	164	31285	YES
a	38.25	6	u	g	k	v	1	t	f	0	t	g	0	0	YES
b	48.08	6.04	u	g	k	v	0.04	f	f	0	f	g	0	2690	YES
a	45.83	10.5	u	g	q	v	5	t	t	7	t	g	0	0	YES
b	36.67	4.415	y	p	k	v	0.25	t	t	10	t	g	320	0	YES
b	56.58	18.5	u	g	d	bb	15	t	t	17	t	g	0	0	YES
b	57.42	8.5	u	g	e	h	7	t	t	3	f	g	0	0	YES
b	42.08	1.04	u	g	w	v	5	t	t	6	t	g	500	10000	YES
b	29.25	14.79	u	g	aa	v	5.04	t	t	5	t	g	168	0	YES
b	42	9.79	u	g	x	h	7.96	t	t	8	f	g	0	0	YES
b	49.5	7.585	u	g	i	bb	7.585	t	t	15	t	g	0	5000	YES
a	36.75	5.125	u	g	e	v	5	t	f	0	t	g	0	4000	YES
a	22.58	10.75	u	g	q	v	0.415	t	t	5	t	g	0	560	YES
b	27.83	1.5	u	g	w	v	2	t	t	11	t	g	434	35	YES
b	27.25	1.585	u	g	cc	h	1.835	t	t	12	t	g	583	713	YES

Iris Dataset

Sepal Len	Sepal Wid	Petal Len	Petal Wid	Class
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
5	3.6	1.4	0.2	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.8	4	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
6.6	2.9	4.6	1.3	versicolor
6.2	2.2	4.5	1.5	versicolor
5.6	2.5	3.9	1.1	versicolor
6.3	3.3	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3	5.9	2.1	virginica
6.3	2.9	5.6	1.8	virginica
6.5	3	5.8	2.2	virginica
7.6	3	6.6	2.1	virginica
4.9	2.5	4.5	1.7	virginica
7.3	2.9	6.3	1.8	virginica
6.4	3.2	5.3	2.3	virginica
6.5	3	5.5	1.8	virginica
7.7	3.8	6.7	2.2	virginica
7.7	2.6	6.9	2.3	virginica
6.9	3.2	5.7	2.3	virginica
5.6	2.8	4.9	2	virginica
7.7	2.8	6.7	2	virginica

Appendix B: Conference Papers

Paper 1: Generating Synthetic Datasets for Experimental Validation of Fraud Detection

Generating Synthetic Datasets for Experimental Validation of Fraud Detection

Ikram Ul Haq, Iqbal Gondal, Peter Vamplew, Robert Layton
ICSL, Faculty of Science and Technology, Federation University, Australia
PO Box 663, Ballarat 3353, Victoria
ikramulhaq@students.federation.edu.au, {iqbal.gondal, p.vamplew}@federation.edu.au, robertlayton@gmail.com

Abstract
Frauds are dramatically increasing every year, resulting in billions of dollars in losses around the globe mainly to banks. One of the key limitations in advancing research in the area of fraud detection is the unwillingness of banks to share statistics and datasets about this fraud to the public due to privacy concerns. To overcome these shortcomings, in this paper an innovative technique to generate highly correlated rule based uniformly distributed synthetic data (HCRUD) has been presented. This technique allows the generation of synthetic datasets of any size replicating the characteristics of the limited available actual fraud data, thereby supporting further research in fraud detection. The technique uses reference data to produce its characteristic measures in terms of Ripple Down Rules (RDR) ruleset, classification and probability distribution to generate synthetic data having same characteristics as reference data. In the generated data, we have ensured that the distribution of individual and the combination of correlated attributes is maintained as per reference data. Further, the similarity of generated data with reference data is validated in terms of classification accuracy using four well-known classification techniques (C4.5, RDR, Naïve Bayes and RandomForest). Instance-based learning classification techniques were used to validate the classification accuracy further as instance-based learners classify the instances to the nearest neighbour instances using similarity functions. The empirical evaluation shows that the generated data preserves a high level of accuracy and data distributions of single attributes as well as the combination of attributes and is very similar to the original reference data.

Keywords: Synthetic Data Generation, Fraud Analysis, Classification, Rule based, Uniformly distributed, Ripple Down Rules

1 Introduction

Online banking frauds are resulting in billions of dollars losses to the banks around the world. In 2008, Phishing related Internet banking frauds costed banks more than US$3 billion globally (McCombie, 2008). Microsoft Computing Safety Index (MCSI) survey (2014) has highlighted that the annual worldwide impact of phishing and various forms of identity theft could be as high as US$5 billion and the cost of repairing damage to peoples' online reputation is much higher at around US$6 billion, or an estimated average of US$632 per loss (Marican & Lim, 2014). Fraud detection for online banking is a very important research area but there are a number of challenges facing research on this topic. In particular knowledge on banks' fraud detection mechanism is very limited and banks do not publish statistics of the fraud detection systems (Maruatona, 2013). Most of the security is provided by third party IT-companies who also protect their intellectual property from their competitors. So both banks and IT security companies do not publish information on their security systems. Bolton & Hand (Bolton & Hand, 2002) also highlight that development of new fraud detection methods is difficult because the exchange of ideas in fraud detection is very limited but authors also support the notion that fraud detection techniques should not be described with details publically; otherwise criminals may also access that information.

To conduct innovative research in fraud analysis, a large amount of data is required. Banks do provide data in some cases, but the data is normally either in small volume or may not provide specific features which are needed to verify new research techniques and algorithms. With the consideration of these limitations, a viable alternative is to generate synthetic data. This paper presents an innovative technique for generating simulated online banking transaction data and evaluates how well this simulated data matches the original, small set of reference data. Further, paper presents fraud detection study on the synthetic data.

Synthetic data can be used in several areas and benefits of synthetic data is well presented by (Bergmann, n.d.):

- It allows controlling the data distributions used for testing. So the behaviour of the algorithms under different conditions can be studied.
- It can help in performance comparison among the different algorithms regarding the scalability of the algorithms.
- It creates instances having the finest level of granularity in each attribute, whereas in publicly

available datasets anonymization procedures are applied due to privacy constraints.

2 Related Work

The idea of synthetic data generation is not new, as in 1993, Donald B. Rubin generated data to synthesize the Decennial Census long form responses for the short form households (Rubin, 1993). However, it has not been applied to the area of online banking fraud.

Various attempts have been made to generate synthetic datasets. One technique uses uni-modal cluster interpolation e.g. Singular value decomposition interpolation (SVDI) (Coyle, et al., 2013). This technique presents a method that uses data clusters at certain operating conditions where data is collected to estimate the data clusters at other operating conditions, thus enabling classification. This approach is applied to the empirical data involving vibration-based terrain classification for an autonomous robot using a feature vector having 300 dimensions, to show that these estimated data clusters are more effective for classification purposes than known data clusters that correspond to different operating conditions. SVDI's main shortcoming is that the estimates of data clusters and known data clusters have the same number of samples.

Different frameworks to synthesise the data (Anon., 2015), (Bergmann, n.d.), (Anon., 2015), (Maj, 2015) have been studied but all of these frameworks neither classify the data nor are based on any existing datasets. One attempt was to generate synthetic census based micro-data with customization and using extensibility of an open-source Java based system (Ayala-Rivera, et al., 2013). In data generation process, they used probability weights by capturing frequency distributions of multiple attributes. Due to attribute interdependency, they also applied attributes constraints, but they have not applied the weightage on the combination of attributes. It might be possible that distribution of individual attributes is same in the generated data as in the reference, but this distribution cannot be guaranteed for the combination of the attributes. The generated data, cannot be used in the domain of classification problems, as this is not the classified data. Another attempt was made to generate constraint-based automatic test data. The technique is based on mutation analysis and creates test data that approximates relative adequacy (DeMilli & Offutt, 1991). This technique is used to generate test data for unit and module testing. This paper does not mention whether this technique is also applicable to produce data for classification.

Chawla et al present synthetic minority over-sampling technique, which is based on the construction of classifiers from imbalanced datasets. Under-sampling of the majority (normal) class has been proposed as a good means of increasing the sensitivity of a classifier to the minority class. Their method of over-sampling the minority class involves creating synthetic minority class examples (Chawla, et al., 2002). This approach is ideal for imbalanced data where the requirement is to reduce majority class and increase the minority class. This technique is not ideal to increase overall data size.

In another paper (Yoo & Harman, 2012) have proposed a technique to generate additional test data from existing reference data. Their paper highlights that mostly existing automated test data generation techniques tend to start from scratch, implicitly assuming that no pre-existing test data is available. However, this assumption may not always hold, and where it does not, there may be a missed opportunity; perhaps the pre-existing test cases could be used to assist the automated generation of additional test cases. They have used search-based test data regeneration technique; that can generate additional test data from existing test data using a meta-heuristic search algorithm (Yoo & Harman, 2012). But the generated data, cannot be utilized in the domain of classification problems, as it does not have classification labels.

Another synthetic data generation and corruption technique is by Christen & Vatsalan (Christen & Vatsalan, 2013), which generates data based on real data having the capability to produce data for Unicode character sets as well. This technique also caters attribute distribution and dependency. Besides these features, this technique is lacking labelled data and attribute distribution multiple attributes. One novel technique is to generate synthetic data for electronic medical records proposed by Buczak et al (Buczak, et al., 2010). However, this technique can generate data mainly for medical domain having the laboratory, radiology orders, results, clinical activity and prescription orders data elements.

In this paper, an innovative technique has been presented which generates highly correlated rule based uniformly distributed synthetic data for fraud analysis. Empirical results are presented by comparing the generated data and original reference data. We have compared the distribution of individual attributes and combinations of correlated attributes. Classification accuracy results for fraud detection are also observed with four well-known classification techniques. The empirical results show that the synthetic data preserves similar characteristics as the original reference data and have similar fraud detection accuracy.

Knowledge-based systems can represent knowledge with tools and rules rather than via code. Mostly current used fraud detection systems, use knowledge base in their architecture with rule-based as commonly used approach. Ripple Down Rules (RDR) was suggested by Compton & Jansen (Compton & Jansen, 1990) as a solution of maintenance and knowledge acquisition (KA) issues in knowledge-based systems(KBS). Ripple Down Rules (RDR) is an approach to knowledge acquisition. RDR has notable advantages over conventional rule-bases; including, better, faster and less costly rule addition and maintenance approaches. Another benefit is the addition of prudence analysis of RDR systems which allows the system to detect when a current case is beyond the system's expertise by issuing a warning for the case to be investigated by the human. Prudence Analysis (PA) was introduced by Compton et al (Compton, et al., 1996).

The synthetic data generation approach can be used to generate data for any classification domain, but in this

paper, test data has been generated to simulate bank transactions to study fraud analysis in banking domain.

In the remainder of the paper, section 3 presents our methodology in detail, while section 4 presents empirical results to show the working of the proposed technique. Finally, paper is concluded in section 5.

3 Synthetic Data Generation Using Highly Correlated Rule Based Uniformly Distribution

Synthetic data is generated with following desired characteristics:

- In some attributes, the generated values should have constraints due to the attribute interdependency on those attributes.
- The continuous attributes values should be within predefined ranges set in the constraints.
- Single attributes should have similar attribute distributions.
- Paired attributes should have similar attribute distribution as the reference data.
- Data should have classification labels.

A high-level flowchart is given in Figure 1. The process is explained in Algorithm 1.

Step 1	Load Reference data in a two-dimensional matrix using (1)
Step 2	Check attribute interdependency. Calculate attributes and class distributions from Reference Data using (2).
Step 3	Generate the Ruleset
Step 4	Start New Instance
Step 5	Generate attributes values from 1 to n with discrete probability distributions using (4).
Step 6	Validate generated attributes values with the ruleset expressions. (if all expressions are not validated then ignore the instance) GOTO STEP 4
Step 7	If generated attributes are validated in STEP 6 then assign the classification label to these attributes (if not classified ignore the instance) GOTO STEP 4
Step 8	Validate class distribution (if not within range ignore the instance) GOTO STEP 4
Step 9	Finalize the Instance.
Step 10	Repeat from STEP 4 to 9 till required instance count matches.
Step 11	Store Generated Data.
Step 12	END

Algorithm 1

A true representation of a generated synthetic data can be ensured by generating Ripple Down Rules (RDR) from reference data and then generating data samples ensuring the distribution of both individual attributes and combinations of attributes remain the same as in the sample reference dataset. Uniform distribution is applied on the attributes to keep data similarity. An innovative HCRUD technique is proposed in this paper to generate synthetic data with desired characteristics.

Reference data is a two-dimensional matrix as given in (1)

$$D_R = [\, d_{ij} \,] \quad (1)$$

where D_R is reference data and i are the attributes from 1 to n and j are rows from 1 to m.

Due to attribute interdependency in some attributes, constraints are applied to those attributes. The probability distribution of attributes is calculated with the ratio of the instances having a particular attribute value over the total instances in the reference dataset.

$$P_i = | D^i_R | / | D_R | \quad (2)$$

Where Pi is the proportional value of the attribute i and $| D_R |$ is the cardinality of D_R i.e reference data and D^i_R are the instances having attribute i.

In the first step, reference data is loaded from the source in the form of a matrix. Attribute interdependency, attributes and class distributions are calculated in 2nd step. In the 3rd step, rules are generated from the reference data. Instance creation is initiated in 4th step. In step five attributes values are generated by applying attribute interdependency and the discrete probability distribution on single as well as the combination of attributes, which is calculated in step 2. In sixth step an instance is formed with the generated attribute values which are then validated based on the established rules, ensuring single and multiple attributes distributions resemble with reference data. The instance is ignored if the instance is not validated from all the expressions from the ruleset. In step seven, after validating the instance, a classification label is generated for the instance. In step eight, it is ensured that class distribution in the generated datasets is also maintained by ensuring the class distribution is within the threshold values. The instance is also ignored if a particular class distribution exceeds the threshold, calculated in step 2. Instance is finalised in step 9 and the steps 1 to 9 are repeated till the desired instance count is reached. Figure 1 is showing a high-level flowchart of the data generation process.

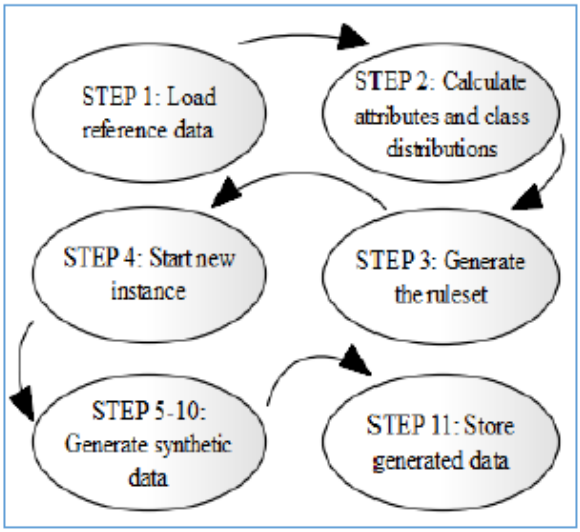


Figure 1: Synthetic Data Generation

The process of generating synthetic data is explained in detail in Figure 2.

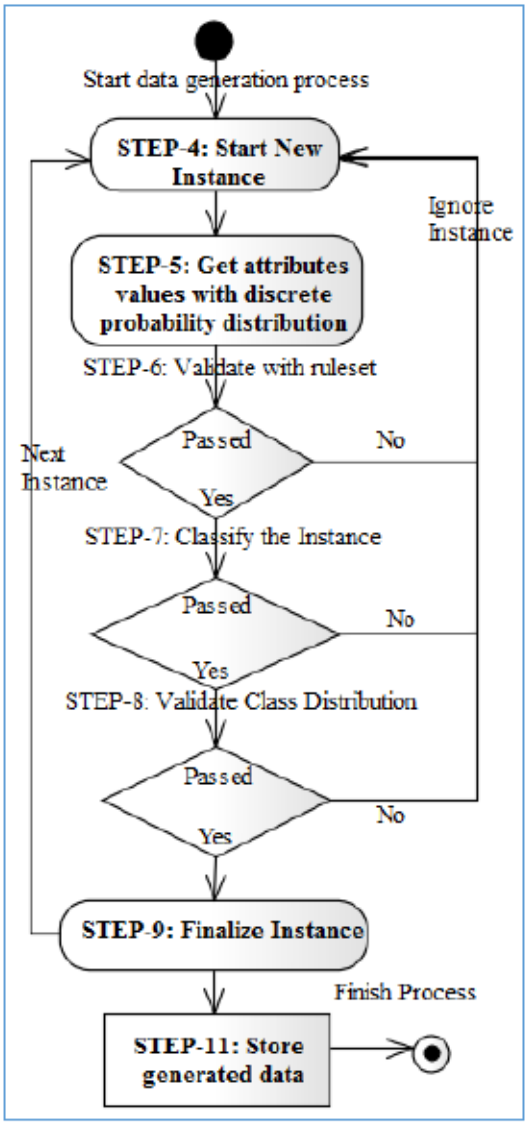


Figure 2: Detailed process to generate data

3.1 Applying HCRUD to Generate a Synthetic Fraud Dataset

To evaluate fraud detection algorithms in the banking data logs, a synthetic data emulating bank transactions has been generated, which is a mix of numerical and alphabetical attributes. An obfuscated dataset of 1775 internet banking transactions from a commercial bank was used to generate synthetic data. Although the dataset is small, the HCRUD technique presented in this paper demonstrates that a synthetic dataset can be generated of any desired size from small reference data. Format and structure of a typical online bank transaction dataset is given in (Maruatona, 2013). The attributes of sample dataset are shown in Table 1. Different banks and fraud detection systems adopt different nomenclatures for transactions.

Name	Description	Type
Transaction ID	Unique ID for transaction	Label
Transaction Type	Type of transaction	Discrete
Account From	Source account	Label
Account To	Destination account	Label
Account Type	Type of account in use	Discrete
Event time	Time of transaction	Time
Session ID	Unique session ID	Label
Browser String	String describing browser	Label
IP Address	IP address for machine	Label
Country	Host country for given IP	Label
Trans Amount	Transfer amount (if Transfer)	Continuou
Biller Code	Unique biller code	Label
Biller Name	BPay Biller business name	Label
Log in ID	User's log in ID	Label
Log in Time	Time of log in	Time
Log in Count	Logins count for the day	Continuou
Password change	Password changes count	Continuou s

Table 1: A sample bank transaction attributes

Discrete probability distribution has been applied on the combination of attributes, i.e. transaction type and class to ensure close resemblance with the sample data:

$$F(x) = P(a \leq x \leq b) = \sum_{k=a}^{b} f(k) \quad (3)$$

where x takes value k between a and b. For combination of the attributes, x is representing the combined value of the paired attributes Transaction Type and Class. Table 2 shows the distribution detail for the combination of attributes.

Transaction Type	Class	Probability
BPAY	Anon	0.022
BPAY	Fraud	0.083
BPAY	Non	0.208
PA	Anon	0.076
PA	Fraud	0.226
PA	Non	0.386

Table 2: Distribution of the attributes for the combination of attributes

Where PA is Pay Anyone and BPAY is a transaction type through which utility bills and other service providers can be directly paid. The class attribute represents the classification of Anon as anonymous and Non as not a fraud. Only one combination of paired attributes is shown as an example here. More paired attributes, even more than two attributes can also be taken, but the more attributes we add, the more would be the ignored instances as mentioned in step 6 in Algorithm 1; hence it will take more time to generate the synthetic dataset. Experimental evaluation has shown that there are about 0.1% to 0.12% ignore cases by taking one combination of paired attributes.

Similarly, discrete probability distribution is applied on individual attributes i.e. transaction type and class separately as shown in (4)

$$\sum_{k=a}^{b} f(k)=1 \quad (4)$$

Table 3 and 4 show the distribution details for single attributes.

Transaction Type	Probability
BPAY	0.313
PA	0.688

Table 3: Single attribute distribution for Transaction Type

Account Type	Probability
Business	0.227
Other	0.001
Personal	0.773

Table 4: Single attribute distribution for Account Type

Sum of the probabilities for both individual attributes is 1.0 Transaction Type and Account Type are the most significant attributes, so distributions detail of these two attributes is discussed above as an example.

3.2 Classification Techniques used for data validation

The system is trained with generated datasets and tested on bank dataset. Datasets of different sizes were generated ranging from 5,000 to 1 million; detail is given in Table 8. Classification accuracy of the generated dataset is observed and compared with four well-known classification techniques, which are Decision Tree (Quinlan, 1993), Ripple Down Rules (Compton & Jansen, 1990) (Richards, 2009), Naïve Bayes (Swain & Sarangi, 2013) and RandomForest (Breiman, 2001).

3.2.1 Instance-Based Learning (IBL)

Aha et al have presented an instance-based learning (IBL) framework which generates classification predictions using only specific instances by applying similarity functions (Aha, et al., 1991). IB1 and IBk are instance-based learners (IBL) (Chilo, et al., 2009) which are also used for testing the classification accuracy in this paper. IB1 is the simplest instance-based learning, nearest neighbour algorithm where similarity function is used. It classifies the instance according to the nearest neighbour identified by Euclidean distance approach (Chilo, et al., 2009) (Aha, et al., 1991). IBk is similar to IB1, but the difference is that in IBk, the K-nearest neighbours are used instead of only one. Three different distance approaches are employed in IBk, including Euclidean, Chebyshev and Manhattan Distance (Chilo, et al., 2009).

3.3 HCRUD Implementation for Data Generation

Weka is well-known data mining tool having a collection of machine learning algorithms and Ridor is a Ripple Down Rules(RDR) implementation in Weka. In this paper, RDR ruleset is generated by using RDR classification from Ridor.

$$R_R = C(D_R) \quad (5)$$

where R_R is set of RDR format ruleset obtained by RDR classification function C. When reference data D_R is classified with Ridor in Weka, it not only classifies the data but also generates a ruleset in RDR format.

A sample format of Ripple Down Rule Learner ruleset is given in Figure 3 that is used in this technique to produce rules from reference data.

```
Except (Browser = Alt) => Class = Fraud  (546.0/0.0) [252.0/0.0]

Except (Network_Count <= 6.5) and (Transfer_Amt > 277.75) =>
Class = Non  (37.0/0.0) [14.0/0.0]

Except (Network_Count <= 11) and (Login_Count > 11.5) and
(Login_Count <= 16.5) => Class = Non  (41.0/1.0) [31.0/1.0]

Except (Source_Acc = Business) and (Network_Count > 8) =>
Class = Non  (4.0/0.0) [1.0/0.0]

Except (Network_Count <= 2.5) and (Acc_Type = PA) and
(LogTime = PM) and (Network_Count > 1.5) => Class = Fraud
(28.0/4.0) [7.0/1.0]
```

Figure 3: A sample of an RDR Ruleset

JEXL name stands for Java Expression Language, an implementation of Unified EL(Expression Language) (Foundation, 2015), JEXL is used to get advantage of extra operators which is used in the rules compactness and to facilitate the implementation of dynamic and scripting features in this technique. The ruleset is transformed from RDR format to JEXL format, attributes-distributions and weightage calculated from reference data is fed to the proposed technique to generate the synthetic data. Figure 1 shows the abstract representation of the technique, while Figure 2 shows the detailed working of the synthetic data generation process. For compactness and efficiency, the generated rules are transformed to (JEXL) format:

$$R_J = T(R_R) \quad (6)$$

where R_J is JEXL format ruleset and R_R is set of RDR rules and T is transformation function of RDR ruleset.

A typical sample of JEXL expressions is shown in Figure 4.

```
Network_Count > 10 & Network_Count <= 12

Transfer_Amt > 2990 & Browser = Moz_4 & Country = AU

Login_Count <= 3 & Country = UK

BPay_Amt > 4750 & Browser = Moz_5Win & Country = AU

Transfer_Amt > 1005.5 & Browser = Opera

Acc_Type = BPAY & Source_Acc = Credit & Browser = Moz_4

PwdChange > 1 & Browser = Moz_5Win
```

Figure 4: JEXL expressions sample

Single classification, JEXL based implementation of RDR is developed and used in this technique to generate class labels to each generated instance. HCRUD generates dataset in variety of formats including Comma separated values (CSV), LibSVM and Attribute-Relation File Format (ARFF), which are widely used data formats in any data mining and machine learning tools. A comma separated values (CSV) format is shown in Table 5 as an example.

Transaction Type	Amount	Account type	Login Count	Network Count	Pwd Changes	Login Time	Browser String	Country	Class
PA	4,000	Other	1	1	1	AM	Alt	Other	Non
BPAY	1,200	Personal	6	3	0	AM	Alt	Other	Non
PA	3,000	Business	1	1	0	AM	Moz_4	AU	Fraud
PA	4,000	Personal	7	1	0	AM	Alt	Other	Fraud
BPAY	860	Personal	3	3	0	AM	Opera	AU	Non
PA	1,500	Personal	14	3	3	AM	Moz_4	AU	Fraud
PA	1,422	Personal	13	2	0	AM	Alt	Other	Non

Table 5: Example Dataset

4 Results

After generating the datasets, the next step was to compare it with original reference data as a benchmark using two different measures. One of the measures was to check the attribute distributions in the reference and generated datasets. Distributions of individual as well as the combination of correlated attributes were also verified, including class association. The second measure was to check the classification accuracy in terms of fraud detection by loading the generated data as training data and reference data as test data. Classification accuracy is verified in Weka with four well-known classification techniques including C4.5/J48, RDR/RIDOR, RandomForest and Naïve Bayes. Instance-based learning classification algorithms (IB1 and IBk) were also used to further verify the classification accuracy outcomes.

4.1 Quality Metric for Attribute Distribution

Root mean squared error (RMSE) is used as a quality measurement indicator, by taking the square root of the mean of the square of all of the errors for data distributions for individual and the combination of attributes. It is represented in (7).

$$RMSE = \sqrt{\frac{1}{N}\sum(D_R - D_G)^2} \qquad (7)$$

Where D_R is reference data and D_G is generated data.

4.1.1 RMSE for Combination of Attributes

RMSE for the distribution of individual attributes as well as combination of attributes were calculated and the experimental evaluation has shown that there is a minor difference in the attribute distribution of reference data and generated data.

The difference in data distribution for the combination of attributes in reference and generated datasets is shown in Table 6.

Transaction Type & Class	Error
BPAY/Anon	0.80
BPAY/Fraud	1.18
BPAY/Non	1.81
PA/Anon	0.80
PA/Fraud	1.22
PA/Non	1.85

Table 6: Error in distribution for the combination of attributes

4.1.2 ..RMSE for Individual Attributes

The difference in data distribution for individual attributes is shown below in Table 7.

Attribute	Value	Error
Class	Anon	0.11
Class	Fraud	0.11
Class	Non	0.00
Transaction Type	BPAY	0.16
Transaction Type	PA	0.22
Account Type	Business	0.03
Account Type	Other	0.03
Account Type	Personal	0.12
Country	AU	0.05
Country	Other	0.11
Browser String	Alt	0.78
Browser String	Mozilla	0.78

Table 7: Error in distribution for single attributes

4.2. Class and Attribute Distributions

Comparisons of the class distribution and distribution of individual as well as the combination of correlated attributes are excellent measures to check how close the generated data is to the original reference data. Fifty datasets were generated and classification and distribution results were averaged and compared with the original reference data.

Figure 5 shows the comparison of distribution by class in generated dataset and in reference dataset; which is very similar.

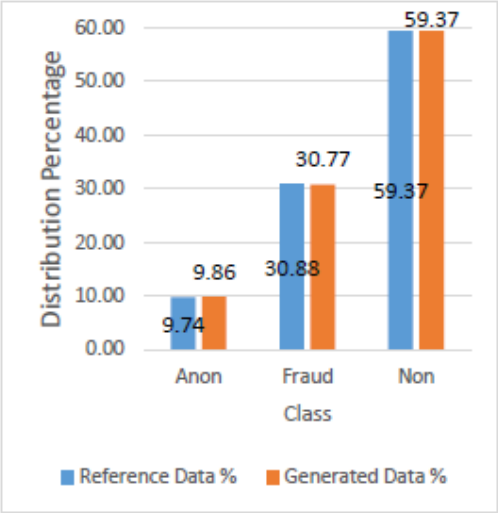


Figure 5: Distribution by class

Figure 6 shows the comparison of the distribution of the combination of attributes (Transaction Type and Class) in generated dataset and in the reference dataset. The results show that the percentages of values from both datasets are very close to each other.

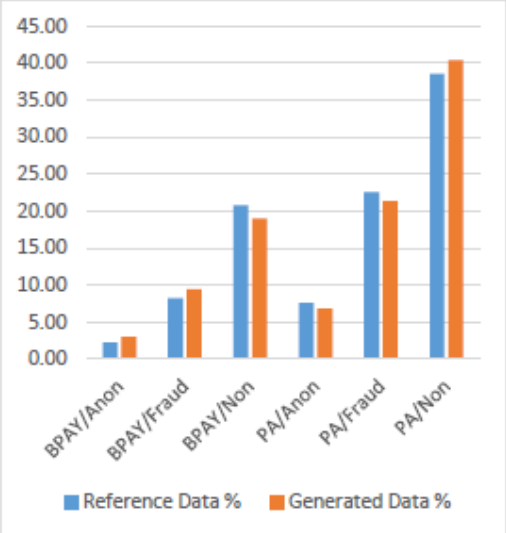


Figure 6: Distribution by Transaction Type and Class

Average time taken to generate instances is also calculated for the individual datasets. Results show that average time taken to generate 1,000 instances is 2.67 seconds. Maintaining attribute and class distributions and assigning class labels to the instance are the few factors, due to which more time is being taken to generate the synthetic datasets. Figure 7 shows the time taken to generate each dataset. It also shows the trend line of time and data size.

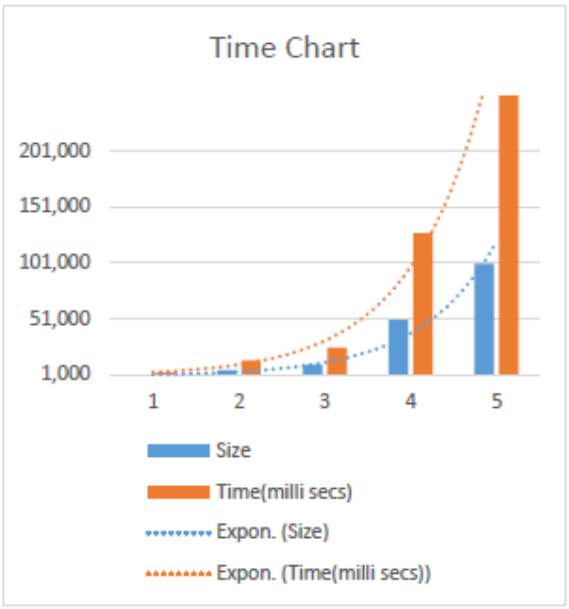


Figure 7: Time taken to generate datasets

4.3 Comparing Classification Accuracy for Fraud Detection

Classification accuracy of the generated dataset is tested with four well-known classification techniques. Table 8, 9 and 10 contain the classification accuracy results; where generated data is used as training data while reference data as test data using C4.5/J48, RDR/RIDOR, Naïve Bayes, and RandomForest classification techniques and instance-based learning (nearest neighbour, similarity based) algorithms as well. The mean classification accuracy for all generated datasets as well as the individual dataset is calculated and is very close to the individual accuracy percentage values.

Dataset	RDR	C4.5	Naïve Bayes	Random Forest	Class. Mean
5k	72.19	75.27	65.70	71.34	71.13
10k	73.96	75.50	65.62	71.74	71.70
25k	76.58	76.24	65.19	75.24	73.31
50k	76.98	76.64	65.41	75.73	73.69
100k	76.98	76.81	65.36	77.09	74.01
500k	77.04	76.98	65.19	77.44	74.10
1mil	76.98	76.92	65.13	77.98	74.22
Dataset Mean	76.03	76.34	65.37	74.93	73.17

Table 8: Fraud Detection classification accuracy results

Classification accuracy results are showing that with the increase of training data (generated data), there is an increase in the accuracy percentage in RDR, C4.5, RandomForest and Classification mean column as well.

Another testing is also performed using cross validation with fold=1755 for both reference and generated data. Fold value of 1755 was taken, due to the reference data size of 1755 instances. Table 9 shows the classification result with four classification techniques with both Reference data and generated data.

Classification	Reference Data	Generated Data	Difference
RDR	77.83	94.02	16.18
C4.5	87.41	96.70	9.29
Naïve Bayes	70.09	89.23	19.15
Random Forest	89.40	94.81	5.41

Table 9: Classification accuracy results with Cross validation

The results are showing that classification accuracy is higher when the system is trained on generated data.

To further verify classification accuracy with instance-based learning (nearest neighbour, similarity based) algorithms, we have performed the evaluation with IB1 and IBk algorithms. Classification accuracy results with instance-based learning are presented in Table 10.

	IBk	IBk	IBk	IB1
Dataset	Euclidean Distance	Chebyshev Distance	Manhattan Distance	
5k	65.64	64.50	66.84	66.95
10k	68.03	67.01	67.12	68.09
25k	71.19	69.18	72.42	72.29
50k	71.69	69.95	73.08	72.89
100k	72.59	70.71	73.73	73.11
500k	73.33	71.28	73.05	73.22
1mil	74.30	73.11	75.44	75.10

Table 10: Classification accuracy results with Instance-Based Learning algorithms

Classification accuracy results shown in Table 8,9 and 10 depict that with the increase of training data (generated data), there is an upward trend of the classification accuracy percentage.

5 Conclusion

To overcome a challenge of limited availability of datasets for fraud analysis studies for financial institutions, an innovative technique: highly correlated rule based uniformly distributed synthetic data has been presented to generate synthetic data. In this paper, we have presented the comparison of the distributions of the original and the synthetic data and the comparison of fraud detection classification accuracy with well-known classification techniques. A single classification, JEXL based Java implementation of RDR is developed and used to generate class labels to each generated instance. In classification accuracy testing, we used generated data as training and original data as test data. Empirical results show that synthetic dataset preserves a high level of accuracy and hence, the correlation with original reference data. Finally, we used an RMSE as a quality metrics for root mean square error to determine the difference of data distribution for individual and the combination of attributes in generated datasets as compared to original reference datasets. Studies have shown very similar distributions of the attributes of generated datasets.

Currently, we are generating the dataset with only 13 attributes of an obfuscated dataset. It needs to be more efficient, otherwise, for high dimensional data, it will take more time. One of the recommended future work is to test this technique on high dimensional data, while another work is to handle missing values from the reference data.

6 References

Aha, D. W., Kibler, D. & Albert, M. K., 1991. Instance-based learning algorithms. *Machine Learning,* 01, 6(1), pp. 37-66.

Anon., 2015. *Generatedata.* [Online] Available at: http://www.generatedata.com/ [Accessed 13 9 2015].

Anon., 2015. *Open Jail - The Jailer Project.* [Online] Available at: http://jailer.sourceforge.net/

Ayala-Rivera, V., McDonagh, P., Cerqueus, T. & Murphy, L., 2013. Synthetic Data Generation using Benerator Tool.

Bergmann, V., n.d. *Databene Benerator.* [Online] Available at: http://databene.org/databene-benerator [Accessed 16 9 2015].

Bolton, R. J. & Hand, D. J., 2002. Statistical Fraud Detection: A Review. *Statistical Science,* 17(3), pp. 235-254.

Breiman, L., 2001. Random Forests. *Machine Learning,* pp. 5-32.

Buczak, A. L., Babin, S. & Moniz, L., 2010. Data-driven approach for creating synthetic electronic medical records. *BMC Medical Informatics and Decision Making,* 10(1), pp. 59-59.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P., 2002. SMOTE: Synthetic Minority Over-sampling Technique. pp. 321-357.

Chilo, J., Horvath, G., Lindblad, T. & Olsson, R., 2009. *Electronic Nose Ovarian Carcinoma Diagnosis Based on Machine Learning Algorithms.* s.l., Springer, pp. 13-23.

Christen, P. & Vatsalan, D., 2013. *Flexible and extensible generation and corruption of personal data.* s.l., ACM, pp. 1165-1168.

Compton, P. & Jansen, R., 1990. *Knowledge in context: a strategy for expert system maintenance.* Berlin Heidelberg, s.n.

Compton, P., Preston, P., Edwards, G. & Kang, B., 1996. *Knowledge based systems that have some idea of their limits.* Sydney, s.n.

Coyle, E. J., Roberts, R. G., Collins, E. G. & Barbu, A., 2013. Synthetic data generation for classification via uni-modal cluster interpolation. *Autonomous Robots,* pp. 27-45.

DeMilli, R. A. & Offutt, A. J., 1991. Constraint-based automatic test data generation. *Software Engineering, IEEE Transactions on,* pp. 900-910.

Foundation, T. A. S., 2015. *Java Expression Language (JEXL).* [Online]
Available at:
http://commons.apache.org/proper/commons-jexl/

Maj, P., 2015. *DBMonster Core.* [Online]
Available at: http://dbmonster.sourceforge.net/

Marican, L. & Lim, S., 2014. *Microsoft Consumer Safety Index reveals impact of poor online safety behaviours in Singapore.* [Online]
Available at: https://news.microsoft.com/en-sg/2014/02/11/microsoft-consumer-safety-index-reveals-impact-of-poor-online-safety-behaviours-in-singapore/

Maruatona, O., 2013. *Internet banking fraud detection using prudent analysis,* Ballarat: University of Ballarat.

McCombie, S., 2008. *Trouble in Florida, The Genesis of Phishing attacks on Australian Banks.* Perth, s.n.

Quinlan, J. R., 1993. *C4.5 : programs for machine learning.* San Mateo, Calif.: Morgan Kaufmann Publishers.

Richards, D., 2009. Two decades of ripple down rules research. *The Knowledge Engineering Review,* 24(02), pp. 159-184.

Rubin, D. B., 1993. Statistical Disclosure Limitation. *Journal of Official Statistics,* 9(2), pp. 461-468.

Swain, S. & Sarangi, S. S., 2013. Study of Various Classification Algorithms using Data Mining. *International Journal of Advanced Research in,* 2(2), pp. 110-114.

Yoo, S. & Harman, M., 2012. Test data regeneration: generating new test data from existing test data. *Software Testing, Verification and Reliability,* pp. 171-201.

Paper 2: Categorical Features Transformation with Compact One-hot Encoder for Fraud Detection in Distributed Environment

Categorical Features Transformation with Compact One-Hot Encoder for Fraud Detection in Distributed Environment

Ikram Ul Haq[1(✉)], Iqbal Gondal[1],
Peter Vamplew[1], and Simon Brown[2]

[1] ICSL, School of Science, Engineering and Information Technology,
PO Box 663, Ballarat, VIC 3353, Australia
ikramulhaq@students.federation.edu.au,
{iqbal.gondal,p.vamplew}@federation.edu.au
[2] Westpac Bank, Melbourne, Australia
simonbrown@westpac.com.au

Abstract. Fraud detection for online banking is an important research area, but one of the challenges is the heterogeneous nature of transactions data i.e. a combination of numeric as well as mixed attributes. Usually, numeric format data gives better performance for classification, regression and clustering algorithms. However, many machine learning problems have categorical, or nominal features, rather than numeric features only. In addition, some machine learning platforms such as Apache Spark accept numeric data only. One-hot Encoding (OHE) is a widely used approach for transforming categorical features to numerical features in traditional data mining tasks. The one-hot approach has some challenges as well: the sparseness of the transformed data and that the distinct values of an attribute are not always known in advance. Other than the model accuracy, compactness of machine learning models is equally important due to growing memory and storage needs. This paper presents an innovative technique to transform categorical features to numeric features by compacting sparse data even if all the distinct values are not known. The transformed data can be used for the development of fraud detection systems. The accuracy of the results has been validated on synthetic and real bank fraud data and a publicly available anomaly detection (KDD-99) dataset on a multi-node data cluster.

Keywords: One-hot Encoder · Compactness · Categorical data · Distributed computing · Hadoop · HDFS · Spark · Machine learning · Sparse data

1 Introduction

Outlier detection techniques have been in use for many applications including Intrusion and Fraud Detection [1–5]. Most of the outlier detection methods use homogeneous datasets having the single type of attributes like numerical or categorical attributes, but real-world datasets often have a combination of these attribute types [6]. For example, Maruatona [4] explains that a typical bank transaction datasets have attributes which are a combination of numeric and categorical attributes.

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 69–80, 2019.
https://doi.org/10.1007/978-981-13-6661-1_6

Numeric features give better performance in classification and regression algorithms. Similarly, clustering algorithms work effectively on the data where all attributes are either numeric or categorical data, as most of the algorithms perform poorly on mixed data types [7]. Huang [8] describes in his finding that clustering methods like k-means are efficient for processing large datasets, but these methods are often limited to numeric data. In addition, machine learning software may only support certain types of data. For example, Apache Spark [9–11] is a highly scalable platform to run machine learning algorithms in a distributed environment, but it accepts only numeric data for classification, regression and clustering algorithms. Therefore, there may be a need to convert categorical variables to a numerical encoding.

Categorical variables are commonly encoded using One-hot Encoding (OHE). Chen [12] indicates that in many traditional data mining tasks, OHE is widely used for converting categorical features to numerical features. OHE transforms a single variable with n observations and d distinct values, to d binary variables with n observations each. Each observation indicates the presence 1 or absence 0 of dth binary variable. However, data becomes sparse after this transformation.

Sparse datasets are common in the big data, where the sparsity comes from factors i.e. feature transformation (OHE), large feature space and missing data [13]. For a given attribute, OHE will increase the number of attributes from one to n distinct values in that attribute, which will not only make the datasets high dimensional but also increase datasets size. Chen [12] believes that other than the accuracy, due to growing memory and storage consumption, compactness of machine learning models will become equally important in the future.

We have presented a technique to transform categorical attributes to numeric attributes and compact the sparsity. The transformed data can be used for the experimental validation and development of fraud detection technique, especially for scalable and distributed data. This technique is tested on a fraud detection bank data and on an anomaly detection KDD-99 dataset, which is widely used as one of the few publicly available datasets for anomaly detection [14]. Multi-node Hadoop cluster is used for experiments, and the performance comparison of the technique has been presented with different classification techniques.

1.1 Contribution

Considering model accuracy and importance of growing memory and storage needs, we have developed a technique to transform categorical attributes to numeric attributes and compact the sparsity as well. An innovative technique is developed and presented in this paper to transform categorical features to numeric features by compacting sparse data even when all the distinct values are not known in advance. Two further models are also developed in One-hot Encoding Extended Compact technique and classification accuracy is evaluated with both models.

Our main contributions in this research are summarized as follows:

(a) Developing One-hot Encoded Extended (OHE-E) technique.
(b) Extending One-hot Encoded Extended with Compactness (OHE-EC).
(c) Develop two further models: First Come First Serve (FCFS) and High Distribution First (HDF) in One-hot Encoded Extended Compact (OHE-EC).

(d) Evaluating classification accuracy, the effect on data size and efficiency in terms of training model and prediction with well-known classification techniques.
(e) Empirical evaluation with a synthetic dataset generated from real bank transaction data and the well-known KDD 95 dataset.

2 Related Work

Several efforts have been made in the past to transform categorical attribute to numeric attributes. First attempt and one of the popular way to convert a categorical feature to a numerical is OHE, but this transformation results in high-dimensional sparse data. Jian et al. [15] have transformed categorical data with Coupled Data Embedding (CDE) technique by extending coupling learning methodology by obtaining hierarchical value-to-value cluster couplings. CDE is slower than other embedding methods, thus is not ideal for large data-sets. It is only applied to unsupervised clustering domain. Another categorical data-representation technique was proposed by Qian et al. [16] with an objective of solving the problem of the categorical data not having a clear space structure. They have not addressed the problem of clustering for a mixed dataset. A comparative evaluation of similarity measures for categorical data is done by Boriah et al. [17]. But the evaluation is performed in a specific context of outlier detection, and relative performance of similarity measures is not studied for classification and clustering. Boriah et al. [17] highlight that several books on cluster analysis [18–20] that discuss the problem of determining the similarity between categorical attributes, recommend binary transformation of data for similarity measures.

To overcome these limitations and for better accuracy, we have presented a technique to transform categorical attributes into numeric attributes and compact the sparsity. This data can be used for the experimental validation and development of fraud detection technique, to check scalability in a distributed environment.

3 Methodology

We have further extended Highly correlated rule-based uniformly distributed synthetic data (HCRUD) [21] to generate numeric synthetic data from mixed reference data. Multi-node Hadoop cluster is used for experiments in a distributed environment with a name-node, resource-manager and multiple workers and data-nodes. The complete process of loading data, filtering categorical features, distribution, transformation, and compactness is explained in the algorithm below.

3.1 Algorithm

```
# Load source data and do Feature selection with Singular
Value Decomposition SVD using Eq.(1).
# Filter categorical features only. Distribute data rows
on worker-nodes in distributed environment in multi-node
Hadoop cluster using Eq.(4). Block size and replication
factor is configurable. We have used 64-MB block size and
three replication factor. Distributing data on worker-
nodes gives efficiency with data locality. Process rows
on worker-nodes in parallel and Process each Row.
  a.  Process each Feature
  b.  IF (Feature is Selected and Categorical)
    i.  For each Feature transform with OHE-E adding extra
    feature using Eq.(5).
# Missing value imputation (MVI) is applied with majority
value of a given attribute for selected attributes. The
decision of taking extra attribute is configured in vari-
ous contextual and model-based profiles. It is evaluated
with different measures explained in 3.3.
    ii. Check sparsity of the vector created with the
    transformation step i using Eq.(2), Eq.(3)
    iii. Compact the sparse data values using Eq.(6)
    FOR Feature 1 to n LOOP
      IF feature NON-ZERO AND NOT NULL
        CompactFeature = featureIndex:feature
      ELSE
        SKIP VALUE
      NEXTVALUE
    ENDLOOP
  c.  IF (more features in the row) Goto step-a
# Compact complete Row using compact values from Step a-c
  CompactRow = EMPTY
  FOR CompactFeature 1 to n LOOP
    CompactRow = CompactRow + SPACE + CompactFeature
    NEXTVALUE
  ENDLOOP
  CompactRow = ClassLabel + SPACE + CompactRow
# Map and reduce tasks are used for processing and re-
source manager manages the processing jobs.
# IF (more Row) from any worker-node Goto Step-4 ELSE
FINISH
```

Source data can be represented in a two-dimensional matrix: $D_S = [d_{ij}]$ where D_S is reference data and having i attributes from 1 to n and j are rows from 1 to m. Feature reduction is done using Singular Value Decomposition (SVD which is a well-known method used for dimensionality reduction). SVD factorizes a matrix into three matrices: U, Σ, and V.

$$A = U\Sigma V^T \tag{1}$$

where U is an orthonormal matrix, Σ is a diagonal matrix with non-negative diagonals in descending order, V is an orthonormal matrix and V^T is the conjugate transpose of V. Sparsity of a vector or matrix can be represented as:

$$V^S = \sum\nolimits_{1(k=0)}^{n} / \sum\nolimits_{1}^{n} \tag{2}$$

where sparsity is the ratio of the sum of attributes of a vector V from 1 to n having value k = 0 to the total attribute values. The sparsity can also be represented as (3), which is 1 minus, the sum of the number of attributes which are non-zero.

$$V^S = 1 - \sum\nolimits_{1(m\neq 0)}^{n} \tag{3}$$

where m are the attribute values, which are non-zero.

3.2 Data Blocks

When a file is stored in Hadoop [22] Distributed File System (HDFS), the system breaks it down into an individual blocks set and stores these blocks in multiple slave nodes (worker-nodes) in the Hadoop cluster. Rows division in each data block can be calculated with (4).

$$\text{Rows}^{\text{Block}} = \sum \text{Rows}/\text{WorkerNodes}/\text{DataBlockSize}/\text{RowDataSize} \tag{4}$$

3.3 Transformation with OHE-E

One-hot Encoding Extended (OHE-E) is a technique developed in this paper, which transforms categorical attributes to numeric attributes with an extra attribute. Missing value imputation (MVI) is applied with majority value of a given attribute for selected attributes. Transformation with One-hot Encoding Extended with an extra attribute is explained in (5).

$$E^{\text{ohe}-e} = \text{fTrans}\left(A^d\right) \tag{5}$$

where $E^{\text{ohe}-e}$ is One-hot Encoding Extended (OHE-E) format and A^d is attribute with d predefined distinct values and fTrans is transformation function of OHE-E. fTrans(A^n) function transforms a selected and categorical attribute A with n observations and d

distinct attribute values, to d +1 binary attributes with n observations each. Each observation indicating the 1 as true or 0 as false of the dth + 1 binary variable. The dth + 1 variable will be true if an attribute value is not from the predefined attributes values. The extra attribute is only included if there is a possibility of new values from previously known values. The decision of taking extra attribute is configured in various contextual and model-based profiles. It is evaluated with different measures including; ratio of total d distinct values of an attribute with n observations. Threshold applied in bank dataset is 0.005. Another measure is time-bound attribute values. For example, in a banking application, the types of transactions can be enumerated in advance, but other attributes such as the device or browser being used may continue to exhibit novel values over time as technology changes.

3.4 Compactness with OHE-EC

Transformation with conventional OHE method makes the data sparse, so compactness of data is suggested and applied in this paper. Compactness on sparse data is applied by omitting all zero and empty attributes values in an instance and keeping the remaining attribute values along with the attribute index. Compactness is explained in (6).

$$C^{ohe-ec} = fCompact \int_{i}^{n} Y(X)\, m \neq 0 \tag{6}$$

Where X is E^{ohe-e} format data from (5) and C^{ohe-ec} is the OHE Extended Compact format and fCompact is a function to compact a row y with only selecting attributes from 1 to n on i^{th} index having m value which is non-zero. Empirical evaluation has shown that after compacting data with OHE-EC, size could be 3x smaller from OHE format.

3.5 Sample Datasets Formats

A sample of the mixed datasets is explained by [21], Table 1 shows sample data, in OHE format for categorical attributes; Transaction Type (BPay and PA), Account Type (Credit, Personal), Browser (Alt, Moz4, Browser New) and Country (AU, NZ, Country. New), while Table 2 shows compact OHE format for same data in Table 1. Compacting process is explained in (6).

Table 1. One-hot Encoding extended dataset.

Class	Bpay	PA	Amount	Credit	Personal	Login	Password	Alt	Moz 4	Moz 5	Brows. New	AU	NZ	Count. New
1	0	1	8210	0	1	5	1	0	0	1	0	1	0	0
0	0	1	5124	0	1	4	1	0	0	1	0	1	0	0
2	0	0	2035	0	1	8	2	0	0	0	0	0	0	1

Table 2. Compact data format.

Class	Attributes
1	2:1 3:8210 5:1 6:5 7:1 10:1 12:1
0	1:1 3:5124 4:1 6:4 7:1 9:1 13:1
2	2:1 3:2035 5:1 6:8 7:2 8:1 14:1

First Come First Serve (FCFS) and High Distribution First (HDF) are two models in this technique. (5) explains that OHE transforms a single variable with n observations and d distinct values, to d + 1 binary variables with n observations each. Each observation indicates the presence 1 or absence 0 of the binary variable. Distribution is calculated for a binary variable having the presence in n observations. In FCFS no sorting is done, but in HDF, the attributes are sorted based on the distribution (higher distribution first). FSFS is efficient in training and testing the model, but it has relatively lower classification accuracy. HDF has better classification accuracy but is little slower in training and testing due to the extra overhead of sorting higher distribution attribute values. Empirical evaluation has shown that if lower distribution attributes are excluded then accuracy with HDF further increases as compared with FCFS.

OHE-EC technique not only reduces dataset size, but gives better performance also in terms of classification accuracy and time (especially on hadoop multi-node cluster), and data can also be used in the Classification techniques which use numeric data only.

4 Results

4.1 Synthetic Bank Transaction Dataset

A synthetic dataset based off actual bank transaction data was generated using the HCRUD technique [21]. Comparison of classification accuracy with synthetic generated mixed data (generated by HCRUD), and numeric data (converted by OHE) is shown in Tables 3 and 4 for different classification algorithms. Training and test data split ratio is 70% and 30% respectively and average results are taken.

Table 3. Accuracy with mixed datasets.

Random forests	Decision tree	Naïve bayes	SVM	OneVsRest	Instances in dataset
96.02%	97.55%	63.59%	60.99%	62.79%	10,000
97.77%	98.85%	64.39%	61.01%	62.58%	100,000
97.90%	98.84%	64.07%	61.57%	62.96%	1,000,000

Table 4. Accuracy with numeric datasets with OHE.

Random forests	Decision tree	Naïve bayes	SVM	OneVsRest	Instances in dataset
97.93%	97.76%	64.86%	93.60%	94.12%	10,000
98.82%	98.85%	64.05%	93.04%	93.21%	100,000
98.88%	98.82%	63.95%	93.24%	93.66%	1,000,000

Classification accuracy results shown in Tables 3 and 4 depict that classification accuracy is better with numeric data (OHE) as compared with a mixed dataset. A T-TEST was performed to determine whether classification accuracy in Tables 3 and 4 are likely to have come from the same two underlying populations that have the same mean or those values have any significant difference. T-TEST, results prove that the classification accuracy results have significant differences.

First come first serve (FSFS) and High distributions first (HDF) are two further models developed in One-hot Encoding Extended Compact (OHE-EC) technique. Tables 5 and 6 show a comparison of classification accuracy with these two models.

Table 5. OHE-EC (FCFS).

Random forests	Decision tree	Naïve bayes	Instances in dataset
97.97%	97.67%	64.77%	10,000
98.84%	98.62%	63.98%	100,000
99.02%	98.95%	63.83%	1,000,000

Table 6. OHE-EC (HDF).

Random forests	Decision tree	Naïve bayes	Instances in dataset
98.16%	97.79%	63.29%	10,000
98.92%	98.76%	64.23%	100,000
99.07%	99.07%	63.84%	1,000,000

The classification accuracy results in Tables 5 and 6 suggest that classification accuracy with OHE-EC (HDF) is slightly better than OHE-EC (FSFS). To confirm this a T-TEST was performed on these results. T-TEST results for Random Forests, Decision Tree and Naïve Bayes are 0.6075, 0.5162 and 0.2113 respectively, indicating that the observed differences between OHE-EC (HDF) and OHE-EC (FCFS) with regards to classification accuracy are not statistically significant.

Other than the classification accuracy, one measure was to compare model's training and perdition time with OHE and OHE-EC. Figure 1 shows training and prediction improvement with OHE-EC in terms of the time.

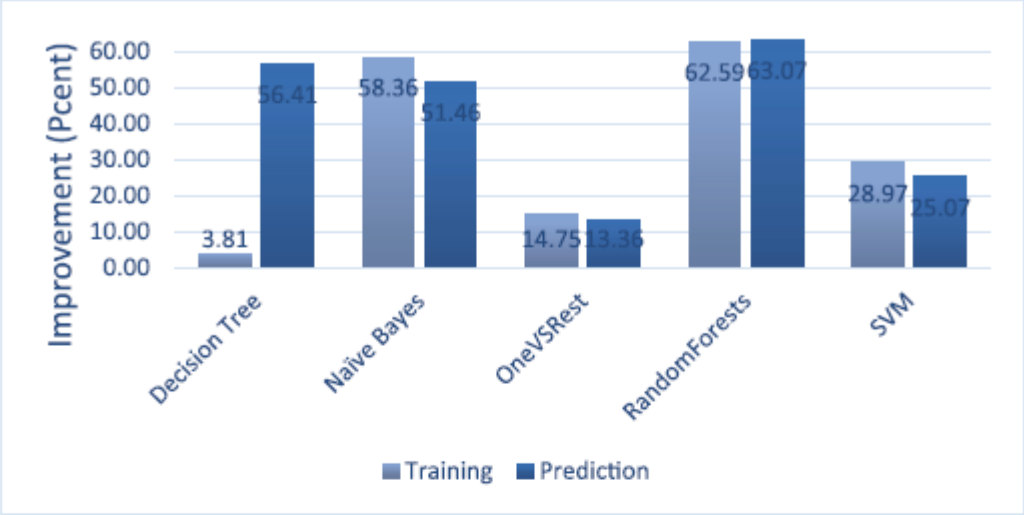


Fig. 1. Average train/prediction time improvement with OHE-EC.

X-axes in the above figure are the classifiers. Y-axes is the average improvement time for different dataset size ranging from very small to large datasets. Results show that there is significant improvement in training and prediction times of the models with OHE-EC. Another empirical evaluation was done with larger datasets only. Figure 2 shows that improvement in prediction time is higher than the training time with larger datasets in almost all classifiers other than Random Forests.

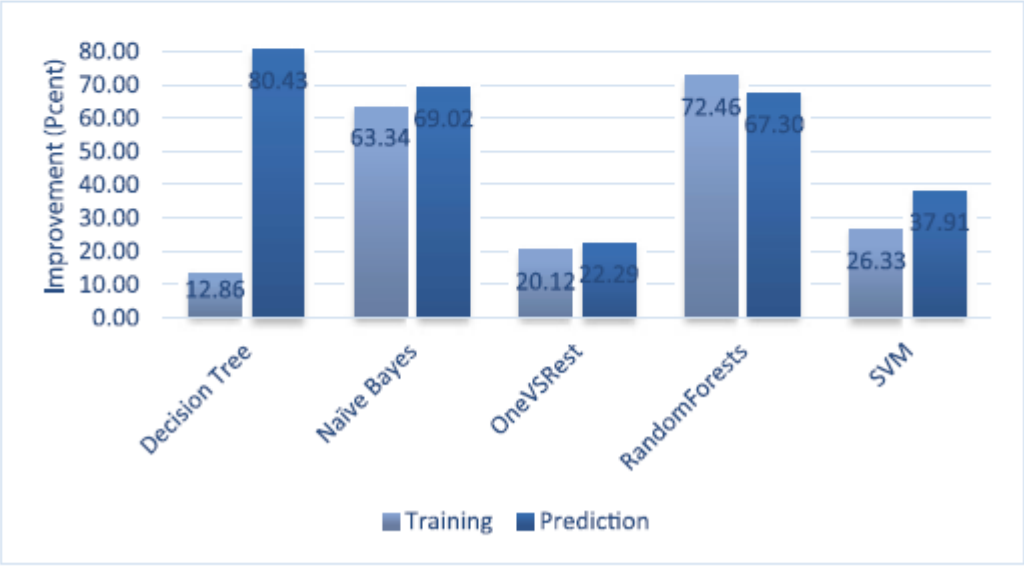


Fig. 2. Large data train/prediction time improvement with OHE-EC.

4.2 KDD Cup Data

The proposed technique was also tested on a KDD-99, a widely used publicly available datasets for anomaly detection [14]. The current datasets contain more than 65 distinct attributes values in service attribute. There is a high possibility that there is new service in the data. One-hot Encoding Extended can transform the row to OHE-E as it is using one extra attribute for new attribute values. Table 7 shows a comparison of classification accuracy with 10 million instances of KDD-99 datasets.

Table 7. Comparison of performance of various classifiers on the KDD-99 dataset.

Random forests	Decision tree	Naïve bayes	SVM	Format	Model
99.973%	99.920%	93.043%	99.991%	Mixed	
99.986%	99.997%	93.711%	99.990%	OHE	
99.99%	99.993%	93.265%	99.997%	OHE-EC	FCFS
99.993%	99.993%	93.463%	99.999%	OHE-EC	HDF

Datasets size of different formats including synthetic data of mixed data and data generated by OHE and OHE-EC were compared. It was observed that datasets size is smallest with OHE-EC, as an average the data in OHE-EC is 3x reduced from OHE. Classification accuracy with OHE-EC with HDF model is also slightly better as compared to the mixed dataset, OHE and OHE-EC (FCFS). Model training and prediction time is also improved with OHE-EC.

5 Conclusion

Fraud detection for online banking is an important area of research, but the heterogeneous nature of data (i.e. mixed data) is challenging. Numeric format data is known to give better performance with classification and some machine learning platforms such as Apache Spark by default only accept numeric data. One-hot Encoding (OHE) is a widely used approach for transforming categorical features to numerical features, but in various datasets, the distinct values of an attribute are not always known in advance. Also, the sparseness of the transformed data is another challenge. Due to growing memory and storage consumption needs; compactness of machine learning models has become much more critical. An innovative technique is presented in this paper to transform categorical features to numeric features by compacting sparse data even when all the distinct values are not known. Results produced by this technique are demonstrated on synthetic and real bank fraud data and anomaly detection KDD-99 datasets on multi-node hadoop cluster. The empirical results show that One-hot Encoding Extended (OHE-E) gives improvements over mixed datasets and One-hot Encoding Extended compact (OHE-EC) not only gives further improvement in reducing the size of datasets, but also an improvement in model's training and prediction time. Two further models OHE-EC (FCFS) and OHE-EC (HDF) are also developed in One-hot Encoding Extended Compact (OHE-EC) technique, where OHE-EC (HDF) gives slightly better classification accuracy as compared to OHE-EC (FCFS).

One of the recommended future work is to test this technique on high dimensional data having and datasets with categorical attributes having a higher number of distinct values.

References

1. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM Sigmod Record (2000)
2. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**(2), 85–126 (2004)
3. Jin, H., Chen, J., He, H., Kelman, C., McAullay, D., O'Keefe, C.M.: Signaling potential adverse drug reactions from administrative health databases. IEEE Trans. Knowl. Data Eng. **22**(6), 839–853 (2010)
4. Maruatona, O.: Internet Banking Fraud Detection Using Prudent Analysis. University of Ballarat, Ballarat (2013)
5. Zhang, Y., Meratnia, N., Havinga, P.: Outlier detection techniques for wireless sensor networks: a survey. IEEE Commun. Surv. Tutor. **12**(2), 159–170 (2010)
6. Zhang, K., Jin, H.: An effective pattern based outlier detection approach for mixed attribute data. In: Li, J. (ed.) AI 2010. LNCS (LNAI), vol. 6464, pp. 122–131. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17432-2_13
7. Shih, M.-Y., Jheng, J.-W., Lai, L.-F.: A two-step method for clustering mixed categroical. Tamkang J. Sci. Eng. **13**(1), 11–19 (2010)
8. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining, (PAKDD) (1997)
9. Pentreath, N.: Machine Learning with Spark, p. 338. Packt Publishing, Birmingham (2015)
10. Meng, X., et al.: Mllib: machine learning in apache spark. J. Mach. Learn. Res. **17**(34), 1–7 (2016)
11. Shanahan, J., Dai, L.: Large scale distributed data science using apache spark. In: 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco (2015)
12. Chen, W.: Learning with Scalability and Compactness, p. 147, Washington (2016)
13. Meng, X.: Sparse data support in MLlib. Apache Spark Community, San Francisco (2014)
14. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications 2009. CISDA 2009, Ottawa, Canada (2009)
15. Jian, S., Cao, L., Pang, G., Lu, K., Gao, H.: Embedding-based representation of categorical data by hierarchical value coupling learning. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (2017)
16. Qian, Y., Li, F., Liang, J., Liu, B., Dang, C.: Space structure and clustering of categorical data. IEEE Trans. Neural Netw. Learn. Syst. **27**(10), 2047–2059 (2016)
17. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: a comparative evaluation. In: Proceedings of the 2008 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics (2008)

18. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1973)
19. Hartigan, J.A.: Cluster Algorithms, vol. 214, p. 1993. Wiley, New York (1975)
20. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, NJ (1988)
21. Ul Haq, I., Gondal, I., Vamplew, P., Layton, R.: Generating synthetic datasets for experimental validation of fraud detection. In: Fourteenth Australasian Data Mining Conference, Canberra, Australia. Conferences in Research and Practice in Information Technology, vol. 170, Canberra (2016)
22. Apache Software Foundation: Apache Hadoop, 26 April 2015. http://hadoop.apache.org/

Paper 3: Enhancing Model Performance for Fraud Detection by Feature Engineering and Compact Unified Expressions

Enhancing Model Performance for Fraud Detection by Feature Engineering and Compact Unified Expressions

Ikram Ul Haq[1], Iqbal Gondal[1], Peter Vamplew[1]

[1]ICSL, School of Science, Engineering and Information Technology, Australia
PO Box 663, Ballarat 3353, Victoria
ikramulhaq@students.federation.edu.au, {iqbal.gondal,
p.vamplew}@federation.edu.au

Abstract. The performance of machine learning models can be improved in a variety of ways including segmentation, treating missing and outlier values, feature engineering, feature selection, multiple algorithms, algorithm tuning/compactness and ensemble methods. Feature engineering and compactness of the model can have a significant impact on the algorithm's performance but usually requires detailed domain knowledge. Accuracy and compactness of machine learning models are equally important for optimal memory and storage needs. The research in this paper focuses on feature engineering and compactness of rulesets. Compactness of the ruleset can make the algorithm more efficient and derivation of new features makes the dataset high dimensional potentially resulting in higher accuracy. We have developed a technique to enhance model's performance with feature engineering and compact unified expressions for dataset of unknown domain using profile models approach. Classification accuracy is compared using well-known classifiers (Decision Tree, Ripple Down Rule and RandomForest). This technique is applied on fraud analysis bank dataset and multiple synthetic bank datasets. Empirical evaluation has shown that not only the ruleset size of training and prediction dataset is reduced but performance is also improved in other performance metrics including classification accuracy. In this paper, the transformed data is used for the experimental validation and development of fraud detection technique, but it can be used in other domains as well especially for scalable and distributed systems.

Keywords: Model Performance, Fraud Detection, Unified Expressions, Feature Engineering, Categorical Data, Compactness, Ruleset, Situated Profiles, RDR.

1 Introduction

The accuracy of a machine learning model can be boosted with the use of various methods such as segmentation [1], adding more data, treating missing [2] and outlier values, feature engineering(FE) [3] [4] [5], feature selection, multiple algorithms, algorithm tuning and ensemble methods. Particularly, feature engineering helps to extract more information from existing data by deriving new features from existing features. It helps to unleash the hidden relationships in a dataset. Derived features may help in explaining

the variance in the training data more accurately and result in higher accuracy. FE could be done using indicator variables, features interaction, feature representation by extracting information from the existing features, transforming categorical to numeric features, by creating dummy features or by using external data. Feature representation can be mainly applied to categorical attributes. In this paper, we have focused on feature representation with minimum knowledge of the domain of an external dataset. One of the challenges in FE is to determine if FE can be applied on a particular feature and whether it could be applied via contextual expressions or via external sources, while another challenge is that data become high dimensional as new features are derived from existing features. We have developed a Feature Engineering and Compact Unified Expressions (FECUE) technique to improve model performance with feature engineering with minimal prior knowledge of the domain of the dataset coupled with compacting the ruleset and dataset with unified expressions using a model-based approach. Performance is measured using three well-known classifiers (Decision Tree [6], Ripple Down Rules(RDR) [7] and RandomForest [8]). The proposed technique is applied to bank datasets. The empirical evaluation has shown that model's performance has improved while training and prediction model sizes have also been reduced. Main contributions are listed below:

- Study of feature engineering and unified expressions to improve fraud analysis.
- Development of feature engineering technique using custom and configurable situated profile models (SPM) when the domain of a dataset is not known in advance.
- Empirical evaluation of the developed technique with multiple datasets.
- Ruleset compactness using contextual expressions and situated profile models.
- Evaluating performance in terms of standard performance metrics including classification accuracy, precision, recall, f-measure, time and ruleset size.

2 Related Work

Some of the known methods of improving model performance are highlighted below:

- Segmentation [1] by dividing the population into several groups.
- Adding more data to produce more accurate models and treating missing [2] and outlier values.
- Feature Engineering [3] [4] [5], extracting more information from existing features.
- Feature selection by finding and the most important subset of features.
- Multiple algorithms by applying a relevant model to see better suitability of models for a particular domain.
- Algorithm tuning by finding optimum parameter values used in the algorithm.

Our research focuses on feature engineering which is being used in different domains to improve model performance. In [3] authors have conducted an educational data mining study; and evaluated feature engineering for KDD Cup 2010 by training the model from students' past behavior and then predicting future performance. Authors in [4] have designed an information extraction technique using feature engineering with a

combination of rule-based and machine learning methods. This technique is applied on narrative clinical discharge summaries. Reid Turner et al. [5] proposed the concepts of FE and evaluated its impact on the software development life cycle. They proposed their research as the first step towards the development of feature engineering and its relationship to other domains. One text classification feature engineering technique is developed by [9], which is ontology guided. This technique utilizes the domain knowledge encoded in the taxonomical structure of the Medical Language System with the help of context-dependent relatedness between pairs of concepts.

These developed techniques have a variety of limitations and are either domain or context-specific. They do not discuss the problem or the solution of the increase of data dimension with the application of FE. Also, the performance impact in terms of either of the classification accuracy, time and model's size is not discussed. FE via external sources is also not used in these techniques. Considering these limitations, we have proposed an innovative technique which improves model performance over a variety of performance metrics. The proposed technique is a situated profile model-based, domain independent FE technique using compact unified expressions.

3 Methodology

Out of various methods available for improving model accuracy, research in this paper focuses on feature engineering and compression of ruleset of the training model. One of the challenges was to identify appropriate FE methods for individual attributes, ideally requiring minimal domain knowledge. Another challenge was the compactness of the ruleset. Four situated profiles models (SPM) are developed and used in this technique to predict features, which type of FE to use and how to apply the ruleset compactness. SPMs are explained in section 3.1. Situated profile models make the technique more generic for different datasets. Consider the nomenclature of a typical bank transaction log as explained by Maruatona [10] Table-7-1.

Categorical attributes represent a type of data which may be divided into groups. Typically, a categorical attribute represents discrete values and have no concept of ordering the values of that attribute. From Maruatona [10] Table-7-1, some of the fields can be used for feature extraction. The developed technique is divided into two parts, feature representation and compactness of the ruleset. A situated profile (SP) [11] defines values relative to the situations, so these are only applied in situations for which they are valid. A situated profile could help in intelligence extraction efficiently. In RDR, the modelling is also based on SPs [10], as it describes every attribute for a particular case. The developed technique is explained in more detail in section 3.4.

3.1 Feature Engineering Techniques for Bank dataset

Many classification algorithms do not use attributes like Event-time, IP Address and Browser string as these type of attributes are ignored in the feature selection process. Feature engineering [12] is a critical and underexplored aspect of building high-quality knowledge base construction systems and is an understudied problem relative to its

importance, especially in fraud detection. One way of FE is extracting information from the existing features, while another way is by using external data sources with some application program interface (APIs) or source like geocoding and demographics. In this paper, we have also applied FE with external data sources.

If we derive new attributes from existing attributes and train the model, we can see that the new attributes are used by the classifier. The newly derived features either can be numeric or can be easily transformed to numeric attributes. Numeric features give better performance in machine learning algorithms. Similarly, clustering algorithms work effectively on the data where all attributes are either numeric or categorical data, as compared to mixed data types [13]. [14] also proved higher classification accuracy with numeric data opposed to mixed datasets. In bank dataset, more attributes can be derived from Event-time, e.g. hour, day, month, year, day-of-week, holiday and weekend-flag. Browser string attribute may further produce attributes like O.S, browser and device identifiers. New attributes derived from an IP Address value could be either four segments separated by token character or location-based attributes. External data sources are available which provide geographic information of an IP Address. These newly derived attributes could also be helpful in identifying suspected transactions in terms of fraud. For example, if event hour is not in normal time, or if it is a holiday or weekend or if the location of the IP Address is different from the actual user's location, then there is higher chance of a potential fraud. Same applies with the attributes derived from Browser string attribute. Different SPMs are formed to aid this method be generic and domain-independent.

3.2 Situated Profile Models

A number of situated profiles models (SPM) were developed to process features and for the ruleset compactness. These models are used for banking dataset, but could also be modified for a specific dataset. Table-1 SPM is a set of tokenizer characters and their applicability to attributes, while Table-2 explains different measures to predict an attribute based on the type and category. With Table-3 FE could be categorized if it can be done via contextual expressions. E.g extracting day-of-week from date field or getting geocoding and demographic information from an IP Address.

Table 1. Tokenizer Character model sample

Token Character	Category	Attribute Index
.	Include	2, 6
	Include	3, 5, 4
;	Include	5
,	Skip	all
)	Skip	5

Table 2. Feature Prediction model sample

Type	Category	Possible values
Attribute Data Type	Comparison	String, Date, Amount, Integer
Tokenizer	Boolean Exists	Yes/No
Tokenizer	Find	Ref: Table-1

Tokenizer	Count	1,2,3
Attribute	Length	0-100

Table 3. FE type model sample

FE Source	Attribute Index
Contextual Expressions	3
Contextual Expressions	4
Contextual Expressions	5

Below table is a sample list of UEL operators, which can be replaced with simple mathematical operator to achieve compactness in UEL ruleset.

Table 4. Rules Compression model sample

UEL Operator	Simple Operator	Types
Between	>=	Integer, Amount
Between	<=	Integer, Amount
Like/In	=	String
Not Between	NA	Integer, Amount
Not In	NA	String

3.3 Challenges and tokenizing a feature value

One of the challenges in FE is how to evaluate which information or features could be extracted from a particular feature, which already exists in the dataset. It cannot be done without domain knowledge or at-least heuristic approach needs to be applied based on the data type. Without domain knowledge of fraud dataset, how we will know that browser OSVer, O.S, Ver and device features can be extracted from raw Browser string. Heuristically, we know that hour, day, month, day-of-week, holiday and weekday flag information can be extracted from a date-time feature and that an IP Address contains geolocation data, which can be extracted by some external APIs.

A new way of FE is introduced in this paper, which can extract information from existing features with a minimum domain knowledge of the dataset. Four situated profile models (SPM) (Table-1 – Table-4) are developed in this technique to predict a feature and to decide the source of feature engineering. This way is explained in Algorithm-1 and in section 3.6 with a rule-based approach. By using this algorithm and the suggested rule-based approach, information can be extracted by tokenizing a feature value with non-alphanumeric characters. E.g comma, space, bracket, colon and semi-colon, Table-1 is configurable to update tokenizer characters with respect to attributes. From a sample date-time value "15/10/2018 23:55:10" six numeric attributes can be extracted by using algorithm-1, which are "15 10 2018 23 55 10". A classifier doesn't need to know which value is an hour, day, month or a year. Similarly from a sample Browser string value "Mozilla/5.0 (iPad; CPU OS 3_2_1 like Mac OS X; en-us) AppleWebKit/531.21 (KHTML, like Gecko) Mobile", O.S, browser and device identifiers

can be extracted. Although the contents of a Browser string will slightly vary based on the browser and the underlying operating system, but once the system knows that it is a Browser string field it can further extract these attributes. A ruleset can be further developed to extract browser name, operating system and the versions, as Browser string contents may vary based on the browser and the O.S. These newly extracted attributes are a combination of categorical and numeric attributes. But the extracted categorical attribute can also be converted to numeric attribute, which was not possible with the original attribute value of Browser string. Various SPMs are developed in this technique for bank dataset, but may also be customized for a particular dataset.

3.4 Algorithms

The developed technique is based on feature engineering and compactness of ruleset for the model. Feature engineering is explained in Algorithm-1, while ruleset compactness is explained in Algorithm-2. Tokenizer characters are maintained in situated profiles for every attribute, as a particular character could be a tokenizer character for one attribute, but not valid for other attributes.

Algorithm-1.

Input: Instance from a dataset. **Output**: Instance with addition of new features with feature engineering.

#Load Source data and perform data cleaning. Do feature selection and filter categorical features and other features having tokenizer characters.

1. Process instances.
2. Process each Feature
3. IF Feature (Is Categorical) or (Having tokenizer characters)

i. Categorise the feature based on Table-1 and Table-2 (explained in more detail in section 3.6.)

ii. For each feature transform and extract new features with FE.

iii. Tokenize / Split with Tokenizer characters from Situated Profiles using Table-1 and Table-2

```
FOR Feature 1 to n LOOP
    IF NEW Tokenizer THEN Update Situated Profiles
    # Situated profiles will manage collection of tokenizer characters on attribute level.
    ELSE IF Tokenizer THEN NewFeatures = ExtractFeatures(feature)
    #Extract feature with the token
    NEXTVALUE
ENDLOOP
```

4. IF (more features in the row) Goto step-2

#Extract features from complete Row from Step 2-4, IF (more Row) Goto Step-1 ELSE FINISH

Algorithm-2.

Input: A unified expression format rule from a ruleset. **Output**: A compact unified expression format rule.

#Load Ruleset.

1. Process each rule in the ruleset and compact the ruleset using fCompact function (1).
2. Process each expression in the rule.
3. IF (Expression is >= or <=) Process current rule and update UEL Rule 3.a

```
    #Update UEL Rule with BETWEEN operator
    ELSE if (Expression is ==)
```

#Process current rule and update UEL Rule 3.a. Update UEL Rule with UEL operators as Table-4

ELSE SKIP

ENDIF

3.a Update Unified Expression Rule (UEL)

#Update with appropriate UEL operator (BETWEEN, IN, NOT IN, LIKE, NOT LIKE) as explained in Table-4 and in section 3.5

4.IF (more expression) Goto step-2

#Process expressions from complete Rule from Step 2-4. IF (more Rules) Goto Step-1 ELSE FINISH

3.5 Unified Expressions Language

In this paper, we have considered rule-based classifiers. One of the well-known classifiers is RDR. We have suggested ruleset compactness in RDR using unified expressions using SPMs. Unified Expressions Language (UEL) can evaluate mathematical expressions with a lot of operators and enables dynamic scripting feature. Some of the advantages of UEL is that it supports more than 30 different operators; and expressions can also invoke functions, which can help in getting external data for feature engineering. For example, extracting geolocation data in bank dataset. Rule-based classifiers use only limited operators. However, using UEL many more operators can be used e.g. IN and LIKE Operators. In FE, features interaction can be achieved by dynamically evaluating expressions using Add, Subtract, Multiply and Divide operators instead of creating new features in the prediction phase. FE with feature interaction will be only needed for training the model. Authors in [14] have highlighted the importance of compactness of the prediction model and demonstrated that a compact prediction model is more efficient. The UEL expression will help in ruleset compactness and will improve performance in terms of the time taken for model prediction.

Algorithm-2 explains compactness with Expression Language using a configurable situated profile model (Table-4). This model uses a relevant UEL operator which can be used based on simple operator and attribute type. Ruleset compactness with unified expressions is explained below:

Rule-1: 'Source_Acc'='Personal' and 'Country'='AU' and Browser='MOZ-5Win' THEN FRAUD

Rule-2: 'Source_Acc'='Personal' and 'Country'='AU' and Browser='MOZ-5Lin' THEN FRAUD

Compressed Rule: (Using IN Operator)

'Source_Acc'='Personal' and 'Country'='AU' and Browser IN ('MOZ-5Lin', 'MOZ-5Win') THEN FRAUD

Other Operator could be BETWEEN for numeric features and LIKE for categorical features.

Compactness of an expression is explained with below equation.

$$R^{comp} = fCompact \int_{i}^{n_V} (expSet)\ m \neq \text{null} \quad (1)$$

Where expSet is a set of expressions from RDR ruleset and R^{comp} is a compact rule set with unified expressions and fCompact is a function to compact an RDR ruleset which compacts simple mathematical expressions from 1 to n from SPM Table-4 on i^{th} rule index having m value which is non-null.

Contextual Expressions

8

Unified expressions can be used to get further useful information from the existing attributes through external sources, e.g. getting geocoding and demographic information from IP Address in bank dataset. Which can help in making further decisions related to fraudulent transactions and will improve model accuracy as well. To make it generic which attributes needs FE from an external source, a situated profile model Table-3 is developed and used in this technique. This model decides FE based on the attributes, which is predicted from two other models Table-1 and Table-2. E.g. Get country information from IP Address may help in detecting suspected tunnel sites usage. We can add a rule when IP Address and user's actual country are different.

Rule: 'Source_Acc' == 'Personal' and 'UserCountry' <> 'IPCountry' THEN FRAUD

3.6 Constructing a feature

Extracting features from the existing feature is a challenging task, especially without knowing the domain of the dataset. However, if we know the feature name in a particular dataset, it will help in extracting more features from this feature. Considering commonly used data types explained by [15], [16] and adding some further measures of feature content length and presence of the token character, a rule-based approach is developed to predict a feature name. To make the technique more generic, four situated profile models are developed and used in this technique. See a ruleset example.

```
Rule-1: DataType='String' and Count(Token_Character='.')=3 THEN IPAddress
Rule-2: DataType ='String' and Token_Character==';' THEN BrowserString
Rule-3: DataType ='String' and (No_Token_Character or Token_Character='_')
THEN SourceAccount
Rule-1, 2 and 3 can also be represented as:
  DataType ='String'
       Count (Token_Character='.') = 3 THEN IPAddress
       Token_Character=';' THEN BrowserString
       (No_Token_Character or Token_Character='_') THEN SourceAccount
```

Comparison with attribute types and checking the existence of a particular and using other measures of length or count is used from the SPMs explained in section 3.1

4 Results

Empirical evaluation was done for both original and the dataset produced by FECUE technique. Performance was measured with a variety of performance metrics including classification accuracy, precision, recall, f-measure, time and ruleset compactness.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

9

$$\text{F-measure} = \frac{2*(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (5)$$

Where TP are correctly predicted positive and TN are correctly predicted negative values, FP when actual class is no and predicted class is yes and FN when actual class is yes but predicted class is no.

4.1 Bank datasets

Various performance metrics with three well-known classifiers has been compared for the original datasets and corresponding datasets with derived attributes after feature engineering using FECUE. The results in Table-5 and Table-6 show that there is an improvement in performance metric results. In this study, 30% and 70% split is done for training and testing datasets. Average measurement was calculated for various dataset sizes ranging from small to large datasets and for multiple simulation runs for each classifier. RIDOR is RDR and J48 is decision tree implementation in WEKA.

Table 5. Performance with Reference Bank dataset

Classifier	Accuracy	Precision	Recall	F-Measure	Time	Ruleset
RIDOR	3.96%	1.85%	4.05%	4.05%	58.06%	26.09%
C45/J48	0.32%	-0.10%	0.00%	0.00%	50.00%	-10.67%
R. Forests	49.39%	91.49%	33.68%	97.39%	-8.33%	

Table 6. Performance with Synthetic Bank dataset

Classifier	Accuracy	Precision	Recall	F-Measure	Time	Ruleset
RIDOR	6.75%	7.34%	6.75%	7.91%	165.32%	50.32%
C45/J48	2.64%	5.87%	6.37%	2.53%	108.41%	15.53%
R. Forests	50.58%	52.42%	50.58%	119.64%	20.26%	

Above tables shows that there is an overall improvement (original and corresponding datasets after FE with FECUE) in all performance metrics with both bank's datasets.

5 Conclusion

Model performance can be improved in a variety of ways including segmentation, treating missing and outlier values, feature engineering, feature selection, multiple algorithms, algorithm tuning and ensemble methods. This paper has presented model accuracy and compactness technique (FECUE), and it is observed that derivation of new features makes the dataset high dimensional. The developed technique has enhanced the model's performance with feature engineering (when the domain of a dataset is not known in advance), with the use of external sources and compact unified expressions.

Multiple situated profile models (SPM) are used to make the technique more generic so that it is applicable on multiple datasets and domains. Performance in terms of classification accuracy, precision, recall, f-measure, time and ruleset compactness is compared using three well-known classifiers. FECUE has been applied on reference bank dataset and multiple synthetic bank datasets. The empirical evaluation has shown that not only the ruleset in training and prediction model are reduced but the performance improvement is also observed in other standard performance metrics. The developed technique is mainly applied in fraud detection area, but it can be used in other domains as well. One of the future works would be to test this technique on a variety of datasets especially with high dimensional data.

References

1. K. Bijak and L. C. Thomas, "Does segmentation always improve model performance in credit scoring?," Expert Systems with Applications, vol. 39, no. 3, 15 2 2012.
2. Z. Xiaofeng, Z. Shichao, J. Zhi, Z. Zili and X. Zhuoming, "Missing Value Estimation for Mixed-Attribute Data Sets," IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 1, pp. 110-121, 2011.
3. H.-F. Yu, H.-Y. Lo, H.-P. Hsieh, J.-K. Lou, T. G. McKenzie, J.-W. Chou, P.-H. Chung, C.-H. Ho, C.-F. C., Y.-H. W., J.-Y. W., E.-S. Yan, C.-W. Ch., T.-T. Kuo, Y.-C. Lo, P. T. C., C. Po and C.-Y. W., "Feature Engineering and Classifier Ensemble for KDD Cup 2010," 2010.
4. Y. Xu, K. Hong, J. Tsujii and E. I.-C. Chang, "Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries," J.A.M.I.A, vol. 19, no. 5, 2012.
5. C. Reid Turner, A. Fuggetta, L. Lavazza and A. L. Wolf, "A conceptual basis for feature engineering," The Journal of Systems & Software, vol. 49, no. 1, pp. 3-15, 1999.
6. J. R. Quinlan, C4.5 : Programs for Machine Learning, Morgan Kaufmann, 1993.
7. P. Compton and R. Jansen, "Knowledge in context: a strategy for expert system maintenance," in Australian Joint Conf. on Artificial intelligence, Berlin Heidelberg, 1988.
8. L. Breiman, "Random Forests," Machine Learning, pp. 5-32, 2001.
9. V. N. Garla and C. Brandt, "Ontology-guided feature engineering for clinical text classification," Journal of biomedical informatics, vol. 45, no. 5, pp. 992-998, 2012.
10. O. O. Maruatona, "Internet banking fraud detection using prudent analysis," University of Ballarat, Ballarat, 2013.
11. M. H. Vastenburg, SitMod: A Tool for Modeling and Communicating Situations, 2004.
12. C. Ré, A. A. Sadeghian, Z. Shan, J. Shin, F. Wang, S. Wu and C. Zhang, "Feature Engineering for Knowledge Base Construction," 2014.
13. M.-Y. Shih, J.-W. Jheng and L.-F. Lai, "A Two-Step Method for Clustering Mixed Categroical," Tamkang Journal of Sci. and Engineering, vol. 13, no. 1, pp. 11-19, 2010.
14. I. Ul Haq, I. Gondal, P. Vamplew and S. Brown, "Categorical Features Transformation with Compact One-hot Encoder for Fraud Detection in Distributed Environment," in The 16th Australasian Data Mining Conference, Bathurst NSW, Australia, 2018.
15. I. H. Witten and F. Eibe, Data Mining: Practical Machine Learning Tools and Techniques, Second Edition ed., M. R. Jim Gray, Ed., San Francisco: Morgan Kaufmann, 2005.
16. B. Durrant, E. Frank, L. Hunt, G. Holmes, M. Mayo, B. Pfahringer, T. Smith and I. Witten, "An ARFF (Attribute-Relation File Format)," University of Waikato, [Online]. Available: https://waikato.github.io/weka-wiki/arff_stable/. [Accessed 9 11 2018].

Paper 4: Unified Expression Ripple Down Rules based Fraud Detection Technique for Scalable Data

Unified Expression Ripple Down Rules based Fraud Detection Technique for Scalable Data

Ikram Ul Haq
ICSL, SEIT, Federation University, Ballarat, VIC, Australia, ikramulhaq@students.federation.edu.au

Iqbal Gondal
ICSL, SEIT, Federation University, Ballarat, VIC, Australia, iqbal.gondal@federation.edu.au

Peter Vamplew
ICSL, SEIT, Federation University, Ballarat, VIC, Australia, p.vamplew@federation.edu.au

ABSTRACT

Fraud detection for online banking is an important research area and higher accuracy is highly desirable. The main challenges in fraud analysis are due to the presence of heterogeneous transactions data and large scale and distributed data. Among existing rule-based techniques for fraud detection, Ripple Down Rules (RDR) is ideal due to its less maintenance and incremental learning. However, banking data sets contains billions of transactions, so the performance of RDR on distributed and Big Data platforms need to be studied for fraud detection applications. A single classification Unified Expression Ripple Down Rules (UE-RDR) fraud deduction technique for Big Data has been proposed and evaluated in this paper. By incorporating the Unified Expressions (UE) into the RDR and evaluating the expressions using the Lift score, the compactness of the ruleset can be achieved and the accuracy of the classification improved. In addition, the paper presents a compact model that fuses Majority and Minority classes for RDR-based classifiers. Classification accuracy is compared with the two existing RDR implementations RIDOR and Integrated Prudence Analysis (IPA) technique and the non-RDR classifier (Naïve Bayes) as well. In order to evaluate the accuracy, this technique has been applied to various datasets: Bank, Synthetic Bank datasets and three publicly available datasets: German Credit, Adult (Census Income) and Credit Approval. Empirical evaluations have shown that not only the ruleset size of training and prediction dataset is reduced, but accuracy of classification is also improved. The results showed an improvement in the classification accuracy when compared to two RDR and non-RDR based classifiers. The proposed technique is used for experimental validation and the development of fraud analysis, but it can also be used in other domains, in particular for scalable and distributed systems.

KEYWORDS

Classification, Fraud Detection, Spark, MapReduce, Hadoop, Machine Learning, Ruleset, Ripple Down Rules, Naïve Bayes, RIDOR, IPA, Unified Expressions.

1 Introduction

Fraud detection for online banking is vital as frauds can affect the core business of the financial industry in terms of loss of confidence of the public in the industry. Online banking frauds are resulting in billions of dollars loss to the banks around the world (McCombie, 2008). As per the Microsoft Computing Safety Index survey (2014), the annual worldwide impact of phishing and various forms of identity theft is about US$5 billion. Internet Crime Complaint Centre has reported a 161% increase in the loses in 2018 (FBI, 2018). Various fraud detection techniques have been developed over the last decade. In view of the importance of fraud detection in the banking sector, higher accuracy of fraud detection techniques is critical. One of the major challenges faced by fraud analysis research is the heterogeneous nature of transactions (Ul Haq, Gondal, Vamplew, & Brown, 2018). Typically, datasets can have both numeric and alphabetical attributes, but numeric data is known to provide better performance for machine learning algorithms. Large-scale data in online banking also requires algorithms to show better performance with scalable and distributed data. (Meng et al., 2016) highlight that Apache Spark is a popular open-source platform for large scale data processing and iterative machine learning tasks.

1.1 Prior Work on Fraud Detection Using Machine Learning

(Kou, Lu, Sirwongwattana, & Huang, 2004) believe that fraud detection research mostly uses data mining, statistics, and artificial intelligence; and fraud is identified from anomalies in data and patterns. (Phua, Lee, Smith, & Gayler, 2010) have surveyed fraud detection research to categorize the research using four main approaches including supervised, hybrid, semi-supervised and unsupervised and; also identified the relationship of fraud detection with other domains. (Melo-Acosta, Duitama-Munoz, & Arias-Londono, 2017) have presented a credit card fraud detection technique using Big Data, but their technique is more specific to imbalance and unlabelled data.

(Herland, Khoshgoftaar, & Bauder, 2018) presented a fraud detection approach for Medicare fraud using three medicare and medicaid services datasets. They use combined dataset for training with three learning methods: Random Forest, Gradient Tree Boosting and Logistic Regression models and used the Area Under the ROC Curve metric to measure the performance of fraud detection. They claim that best fraud detection performance is with the use of the combined dataset. Dataset size is not mentioned, but this technique is not ideal for large datasets, e.g. Synthetic dataset generation based on original seed datasets.

Integrated Prudence Analysis (IPA) is developed by (O. O. Maruatona, 2013) which uses prudence analysis in RDR and has combined two of the previously developed Multiple Classification RDR (RM) and Ripple Down Models (RDM) (Kang, Compton, & Preston, 1995; Prayote, 2007) techniques. A fundamental difference in these techniques is that RM is structural while RDM is attribute-based. The difference in these methods is well explained by (O. Maruatona, Vamplew, & Dazeley, 2012). IPA is a multi-class labels classifier.

1.2 Background to UE-RDR Methodology

RDR is one of the well-known rule-based classification technique and was developed as an alternative to the traditional knowledge-based system (Compton & Jansen, 1988; Kang et al., 1995). (O. O. Maruatona, 2013) acknowledges that RDR is ideal due to its less maintenance and incremental learning capabilities. RDR significantly reduces the time and effort required to make the alteration and ensure the consistency of the rulesets. (Kang et al., 1995; Richards, 2003) have highlighted that RDR systems have been used in many applications and classification domains. RIDOR is an RDR implementation in WEKA and (Compton, 2011) also acknowledges that RIDOR is most widely used RDR machine learner. Figure 1 shows an Iris ruleset generated from RIDOR.

```
class = setosa (150.0/100.0)
Except (petal_len > 2.45) => class = virginica (66.0/0.0) [34.0/0.0]
Except (petal_len <= 4.95) and (petal_wid <= 1.55) => class = versicolor (29.0/0.0) [16.0/0.0]
Except (petal_wid <= 1.75) => class = versicolor (8.0/5.0) [1.0/0.0]
```

Figure 1: Iris RIDOR ruleset.

One of RDR implementation is RIDOR, which also has MapReduce (ASF, 2015) based implementation in WEKA for Apache Hadoop (ASF, 2015) wrapper, which can be used for the classification of large data. However, (Meng et al., 2016; Shanahan & Dai, 2015) highlight that Spark is better as compared to conventional MapReduce. Spark maintains MapReduce's linear scalability and fault tolerance and is nearly 100 times more efficient than MapReduce. Mahout is another machine learning platform for Big Data. (Meng et al., 2016) highlights that Mahout is also based on MapReduce and they observed that Spark's performance and scalability are better than Mahout.

Unified Expressions Language (UEL) is capable of evaluating a number of additional operators that are missing in RDR expressions. Unified Expressions (UE) can also replace existing operators with more efficient operators of IN and LIKE. Using UE, we can prepare compressed rule with a revised Lift score which is the ratio of target response divided by the average response. UEL supports contextual expressions and can also retrieve geocoding and demographics information from fraud datasets (Ul Haq, Gondal, & Vamplew, 2019), that help to filter suspected cases. UE application in the proposed technique is explained in section 2.6. UE can offer a variety of operators that can help with the compactness of ruleset and evaluation

of the expression based on Lift score. Furthermore, the UE can help in choosing the best rules with higher confidence; therefore, the more accurate class label is chosen, which improves accuracy. UE-RDR is implemented on Big Data Spark platform by overcoming the limitation of mixed datasets. Apache Spark performance is known to be better than conventional Apache Hadoop MapReduce (Meng et al., 2016; Shanahan & Dai, 2015) so UE-RDR on Spark will be more efficient than RDR MapReduce based implementation in WEKA and will also have iterative machine learning capability.

UE-RDR fraud detection technique for large scale mixed data has been developed and evaluated in this paper to improve detection accuracy and reduce computation costs. The technique has three models: the minority (UE-RDR-MIN) class, the majority (UE-RDR-MAJ) class-based models and combined model (UE-RDR-MIX). The combined and distinct rules in UE-RDR-MIX model gives better accuracy than the other two models. UE-RDR-MIX is an innovative model and to the best of our knowledge, no study has been on in RDR based classifiers. UE-RDR performance is compared with RDR. The proposed technique is applied to various data datasets (Table 3), including Synthetic Bank datasets and three publicly available datasets from the UCI machine learning repository. Performance is evaluated and compared with two RDR based implementations (RIDOR and IPA) and a non-RDR classifier (Naïve Bayes (Swain & Sarangi, 2013) as well. The empirical evaluation has shown that the model's performance in terms of classification accuracy and ruleset size is better than RIDOR. Classification accuracy with UE-RDR-MIX is better than IPA and Naïve Bayes classifiers.

The main contributions of the paper are listed below:

- Study of UE for RDR and development of a threshold-based approach for ruleset compression with the use of Lift score.
- Development of a single classification Unified Expressions RDR (UE-RDR) technique with three sub-models: UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX. UE-RDR-MIX is an innovative model for RIDOR, which makes use of majority and minority classes and multi-level compactness.
- Empirical evaluation of the developed technique for classification accuracy and ruleset compactness with multiple datasets and comparison with various RDR and non-RDR based classifiers.
- Study of the developed technique on distributed and Big Data machine learning platform, Spark.

In this paper, we are focusing on fraud detection for large scale data and with rule-based classifiers using a supervised approach on labelled datasets. The developed technique can be used on mixed datasets. The developed algorithm is implemented on big and distributed data platform Spark and has shown better accuracy as compared with two of the existing RDR based classifiers and a non-RDR classifier.

2 Methodology

Knowledge-based systems are a major application for concept descriptions. (Littin, 1996) mentions that rules and decision trees are two of the common forms of concept descriptions in machine learning. (O. O. Maruatona, 2013) indicates that commercial banks and financial institutions use approaches like rule-based in their Internet banking fraud detection systems.

2.1 UE-RDR Models

Fraud detection data is a single classification data and UE-RDR is also a single classification model, with UE based on RDR. In UE-RDR technique, three models are developed, UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX. (Littin, 1996) highlights that the inclusion of RDR top-level empty rule is used generally with a default class. (Gaines & Compton, 1995) have used the class that occurs most frequently (Majority) as default in the training data, however in RIDOR by default least frequently used class (Minority) is used as default class. UE-RDR technique is also illustrated graphically as a multi-step process in Figure 2. Figure 3 shows iris ruleset for a UE-RDR-MIN model. But a typical ruleset and a particular rule structure of UE-RDR model is shown below.

{"defaultclass":"CLASS-LABEL", "model":"MODEL-NAME","count":TOTAL-POPULATION,

"rulses":[RULES-COLLECTION]

RULE#

{"number":#,"isParent":true,"level":#,"description":"UE-EXPRESSION","lift":#,"cover":#,"ok":# "class":"CLASS-LABEL","parentid":#, "childrenNodes":#}

2.1.1 UE-RDR-MIN

In this model, least frequently occurring (Minority) class is the default class (like RIDOR), and the rules are for the remaining class labels. i.e. majority class label and other classes. In most of the cases ruleset set for this model is supposed to be larger than the ruleset for UE-RDR-MAJ, as least frequently used class is default class and rules are for the remaining class labels (including majority class).

2.1.2 UE-RDR-MAJ

In this model, most frequently occurring (Majority) class is the default class (as used by (Gaines & Compton, 1995)), and the rules are for the remaining classes. In terms of ruleset size, this model would have a similar size of ruleset as UE-RDR-MIN model.

2.1.3 UE-RDR-MIX

This model is a union of the rules for the minority & majority class models and distinct rules for the remaining class-labels. Rules expressions are further compressed with revised Lift score outlined in sections 2.5 and 2.7. This model is our innovation and does not exist in RIDOR implementation. Algorithms 2a explains this model. In RDR ruleset, one class is the default class and ruleset contain rules for the remaining class labels. We claim that this model gives the best classification accuracy, as shown in Figure 5. Unlike RDR, it contains rules for all class-labels instead of using a default-class. In terms of ruleset compactness, Figure 6 shows that for some datasets, UE-RDR-MIN and UE-RDR-MAJ have good performance as well.

If there are more than two class labels in a dataset, this model also provides a better accuracy for class labels that belong to neither majority nor minority classes. Considering Bank dataset example, the Fraud class label does not fall into majority or minority class, so UE-RDR-MIX model will give better accuracy for Fraud class labels in this dataset. Apart from the overall higher classification accuracy, classification accuracy is also sometimes important for a specific class label. For example, Fraud cases are more important for improved accuracy in the Bank dataset. A wrong prediction of a Fraud case would result in a greater loss compared to the mistake of None or Anon cases. Accuracy results from confusion matrix are shown in Figure 8.

2.2 Algorithms

The developed technique is based on three algorithms. UE-RDR ruleset construction is explained in Algorithm-1, while ruleset compactness is explained in Algorithm-2 and prediction flow with Spark is explained in Algorithm-3. Algorithm-2a is for UE-RDR-MIX model only, which is a further compactness of Majority and Minority class models (UE-RDR-MIN and UE-RDR-MAJ). Figure 2 illustrates UE-RDR process flow and glues three algorithms to demonstrate the three-stages. In Algorithm-3, when a data file is stored in Hadoop (ASF, 2015) Distributed File System (HDFS), the system breaks it down into individual blocks set and stores these blocks in multiple worker-nodes in the cluster. Rows division in each data block can be determined with Eq. (1).

$$Rows^{Block} = \Sigma Rows/SparkNodes/BlockSize/RowDataSize \quad (1)$$

The mentioned algorithms are given below:

ALGORITHM 1: Building Training Model

Input: Ruleset from a RIDOR.

Output: Training model for a UE-RDR.

Begin

1. Process RIDOR ruleset.

2. Process each expression in the ruleset.

3. Get Ok and Cover values of each expression.

4. Calculate Lift score of the expression from Ok and Cover values using Eq. (4).

5. Prepare the expression in UE format using funcUEL Eq. (5).

6. Convert the expression in JSON format with attributes (See Figure 3).

7. IF (more expressions in the ruleset) Goto step-2

ELSE FINISH

End

ALGORITHM 2: Compactness

Input: Training model for a UE-RDR.

Output: Compact UE-RDR Training model.

Begin

1. Process each rule in the ruleset of the training model.

2. Traverse Ruleset & Get Lift score of the rule

2.1. Find merging rule (using the custom thresholds approach listed in Table 2).

2.2. Merge UE rule.

3. Traverse rule to compact UE (See UE operators Table 2)

3.1. Calculate and update revised Lift score, from updated Ok and Cover values of merging rule – see Eq. 4.

3.1 Update UE rule.

3.2 IF (more expressions to process) Goto step-3

3.3 Process all expressions from complete rule from Step 3 – 3.2

4 IF (more rules) Goto Step-1 ELSE FINISH

End

ALGORITHM 2a: UE-RDR-MIX Compactness

Input: Training model for a UE-RDR-MIN and UE-RDR-MAJ.

Output: Compact UE-RDR Training model for UE-RDR-MIX.

Begin

1. Repeat Algorithm-2 with the input of two UE-RDR Training Models.

2. Repeat Steps 1 to 3.2 from Algorithm-2.

End

ALGORITHM 3: Prediction Process

Input: Training model from a UE-RDR and dataset.

Output: Balanced accuracy for the dataset.

Begin

1. Load Dataset

1.1. Process each instance.

1.2. Transform instance to RDD double Vector, including categorical attributes using funcTransRDD Eq. (2).

1.3. Split data on Spark nodes based on the data block size using Eq. (1)

2. Load UE-RDR training model.

3. Load RDD vector collection from data locality.

3.1. Process each rule from the Training Model.

3.2. Transform categorical attributes in expression with funcTransCat function (3).

3.3. Evaluate UE rule expression and pick the predicted class.

3.4. If multiple rules are true then pick predicted class of better Lift score rule.

3.5. IF (more rules in the ruleset) Goto step-3.1

IF (more instances to process) Goto step-3 ELSE FINISH

End

2.2.1 UE-RDR Process Flow

Figure 2 connects three algorithms to illustrate the flow of the three-step algorithms. The dependency in each step and the main and subtasks in each step are clarified there. Loading and Prediction are the two steps in Prediction process.

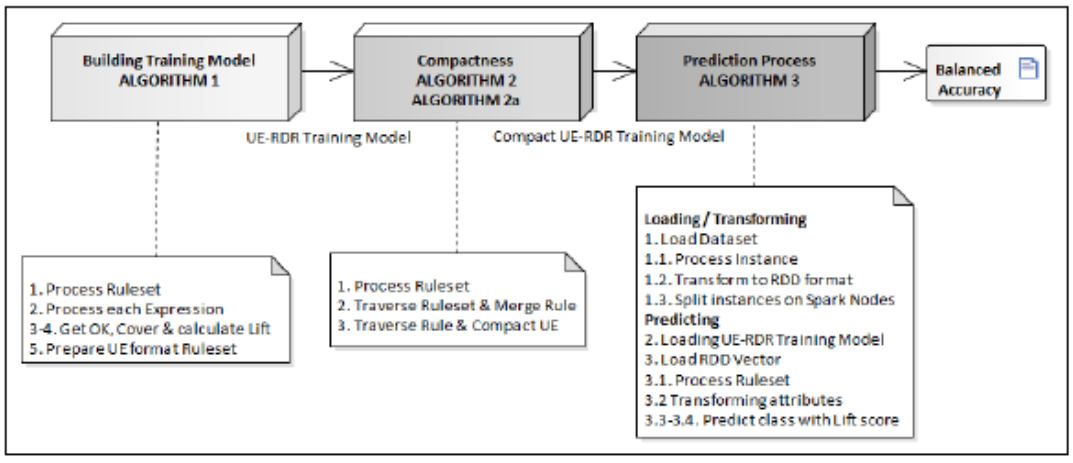


Figure 2: UE-RDR process flow

2.3 Transformations

Due to the large datasets, the developed technique was implemented on Spark. The core of Spark is a concept called the Resilient Distributed Dataset (RDD), which is a collection of records. The default data-format for Spark platform is numeric, however the Bank dataset and many real-life datasets contain mixed attributes. Two transformation functions were developed, which are explained below. The function in Eq. (2) transforms mixed data to numeric RDD format at loading time.

$$\text{Transformation}^{\text{RDD}}=\text{funcTransRDD}\int_{i}^{n_Y} att \neq numeric \qquad (2)$$

where $\text{Transformation}^{\text{RDD}}$ is the RDD format and funcTransRDD is a function to convert a row y with only categorical attributes from 1 to n on ith index.

While function Eq. (3) transforms categorical value of the attribute to numerical value at the expression evaluation time.

$$\text{Transformation}^{\text{CAT}} = \text{funcTransCat}\int_{i}^{n_Y}(att \text{ in exp}) \qquad (3)$$

where $\text{Transformation}^{\text{CAT}}$ is the RDD format and funcTransCat is a function to convert a row y with only categorical attributes from 1 to n on ith index and which exist in an expression. These transformations are necessary in order to evaluate expressions from the original ruleset.

2.4 UE-RDR Ruleset

Figure 3 shows an iris ruleset generated from UE-RDR.

{"defaultclass":"setosa", "model":"UE-RDR-MIN","count":3, "rules":[

{"number":1,"isParent":true,"level":1,"description":"(petal_len > 2.45)","lift":1.5, "cover":100.0, "ok":100.0, "class":"virginica","parentid":0, "childrenNodes":2},

{"number":2,"isParent":false,"level":2,"description":"(petal_len > 2.45) && (petal_len <= 4.95) && (petal_wid <= 1.55)","lift":3.333333, "cover":45.0, "ok":45.0,"class":"versicolor","parentid":1,"isChild":true},

{"number":3,"isParent":false,"level":2,"description":"(petal_len > 2.45) && (petal_wid <= 1.75)", "lift":7.4074, "cover":9.0,"ok":4.0,"class":"versicolor", "parentid":1}]}

Figure 3: Iris UE-RDR ruleset.

Where "Cover" is the number of instances a rule expression correctly identifies and "Ok" is how many instances (out of the Cover) are correctly classified by this rule. While the Lift is the score for Cover, Ok values and the "count" (total population), determined in Eq. (4). While "description" is the rule expression in UEL format.

2.5 Lift

In data mining and association rule learning, the Lift (Martinez, 2019) is a measure of the performance of a model (association rule) for prediction or classification as having an enhanced response (with respect to total population), measured against a random choice of the model. So, Lift is the ratio of target response divided by the average response.

For example, a population has an average response rate of 5%, but a certain model (or rule) has identified a segment with a response rate of 20%. Then that segment would have a Lift of 4.0 (20%/5%). Let us consider Dataset 1 (Bank dataset) with a distribution of transactions from UK with 4 Fraud and 2 None cases, while 4 Fraud cases from AU. Consider the following rule:

Rule: UK implies Fraud, i.e IF Country is UK THEN Class = Fraud

$$\text{Lift} = (\text{Ok} / \text{Cover}) / (\text{Cover} / \text{Total}) \quad (4)$$

The Lift for the rule using Eq. (4) is (4/6)/(6/10) ≈ 1.11

When Country is UK and Class is Fraud = 4 (OK)

When Country is UK = 6 (COVER)

Total population(instances) = 10 (TOTAL)

While evaluating the expressions of the rules, when multiple rules are true, choosing the predicted class of better Lift score (higher confidence) rule will increase accuracy.

2.6 Unified Expressions (UE)

UEL can evaluate mathematical expressions with many operators. It enables dynamic scripting feature. Some of the advantages of UEL is that it supports more than 30 different operators; Rule-based classifiers use only limited operators but using UEL many more operators can be used which are not available in rule-based classifiers, e.g. IN and LIKE Operators. Authors in (Ul Haq et al., 2018) have highlighted the importance of compactness of the prediction model and demonstrated that a compact prediction model is more efficient. The UE will help in ruleset compactness along-with the revised Lift score and hence will improve performance in terms of the time taken for model prediction.

$$\text{Expression}^{UE} = \text{funcUEL}(\text{Expr}^{RDR}) \quad (5)$$

Where Expression is a UE format and Expr^{RDR} is RDR format expression. funcUEL is a function to convert RDR format expression to UEL format. One of the primary functions of funcUEL is to transform RDR operators and operands to UEL operators and operands.

Few of the transformation are:

Transform "and" to "&&" operator

Transform "=" to "==" operand.

To make the transformation more generic, profiles are used for transformation operators and operands. Table 1 shows the transformation detail.

Table 1: RDR and UEL transformation.

RDR	UEL	Category
And	&&	Operator
=	==	Operand

2.7 Compactness

The compactness of ruleset can improve the performance of the algorithms and has been proposed in this paper. One of the challenges was to decide which rules to compact/merge. One of the approaches considered was the nearest neighbor technique using Euclidian based similarity of the instances of two rules. This approach determines (Littin, 1996) distances using Eq. (6) and Eq. (7):

$$Dp = \sqrt{0.2^2 + 0.3^2} = 0.36 \quad (6)$$

$$Dn = \sqrt{0.4^2 + 0.3^2} = 0.5 \quad (7)$$

Where Dp and Dn are the distances of class p and n respectively. But this technique is computationally expensive, so instead, a customized threshold-based approach is used. The measures and the threshold used in the technique are listed in Table 2.

Table 2: RDR and UEL transformation.

Measure	Threshold
Nearest Lift score	<= 0.05
Same parent rule	
Smaller expression rule	<=2
IN / BETWEEN operators	> 2

New values of Ok, Cover and Lift score are calculated for merging rules of customized scheme.

2.8 Experimental Setup

A multimode Hadoop cluster with Spark was set up on a National eResearch Collaboration Tools and Resources (Moloney, Barker, Coddington, & Mecoles, 2011) research cloud to develop and evaluate this technique for large datasets. Spark is ideal for iterative machine learning tasks and is much faster than conventional MapReduce. Figure 4 is a typical diagram of Spark internal execution on a Hadoop cluster, which makes it iterative and more efficient than MapReduce.

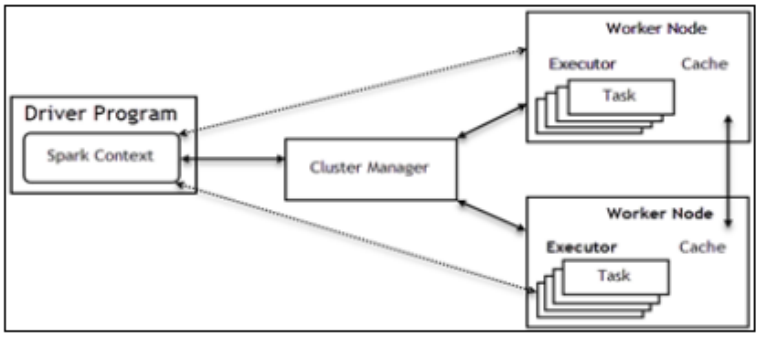


Figure 4: Spark execution flow.

2.9 Dataset characteristics

Characteristics of multiple datasets used for the evaluation are listed in Table 3.

Table 3: Dataset characteristics.

Dataset	Description	Instances	Features
Dataset 1	Reference Bank Data (Ul Haq, Gondal, Vamplew, & Layton, 2016)	1,756	14
Dataset 2	Synthetic Bank Data (Ul Haq et al., 2016)	100,000	14
Dataset 3	German Credit Data (Hofmann, 1994; Prasad & Ramakrishna, 2016)	1,000	11

Dataset 4	Credit Approval (Quinlan, 1987, 1992)	691	16
Dataset 5	Adult (Census Income) (Kou et al., 2004; Zadrozny, 2004)	32,562	8

Synthetic Bank data was generated from reference Bank data using HCRUD (Ul Haq et al., 2016) technique. This technique can produce huge dataset on the Hadoop cluster, which is similar to original reference dataset. The dataset is produced with uniform distribution of class labels, individual and combination of attributes as well. RMSE of the difference of distributions in individual attributes is between 0.00 to .78, while for the combination of attributes is between .80 to 1.85. Spark can use huge datasets, but for evaluation purpose, 100,000 instances of the dataset were used.

3 Results

Classification accuracy of UE-RDR technique is compared with existing RDR implementation in WEKA (RIDOR). An empirical evaluation was performed with various datasets listed in Table 3, with 30% and 70% split for training and testing datasets respectively. Average measurements were taken for various small to large dataset sizes and with five simulation executions. Vertical axes in Figure 5 - Figure 7 are the percentage of performance improvement of UE-RDR models over the other classifiers. Performance comparison for classification accuracy is shown in Figure 5 and Figure 7, where the accuracy is the ratio of correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \qquad (8)$$

Where true positives (TP) are the correctly predicted positive values and true negatives (TN) are the correctly predicted negative values, false positives (FP) when actual class is no and predicted class is yes and false negatives (FN) when actual class is yes but predicted class is no.

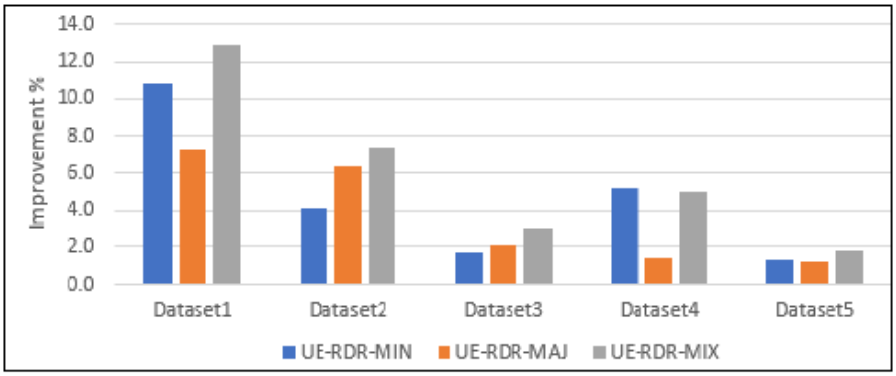


Figure 5: % Improvement in Classification Accuracy over RIDOR.

The results show that classification accuracy with all the datasets is improved. Out of the three UE-RDR models, UE-RDR-MIX performance is best among all datasets other than Dataset 4 (Credit Application dataset) where UE-RDR-MIX and UE-RDR-MIN accuracy is almost the same.

Similarly, ruleset compactness results are displayed in Figure 6.

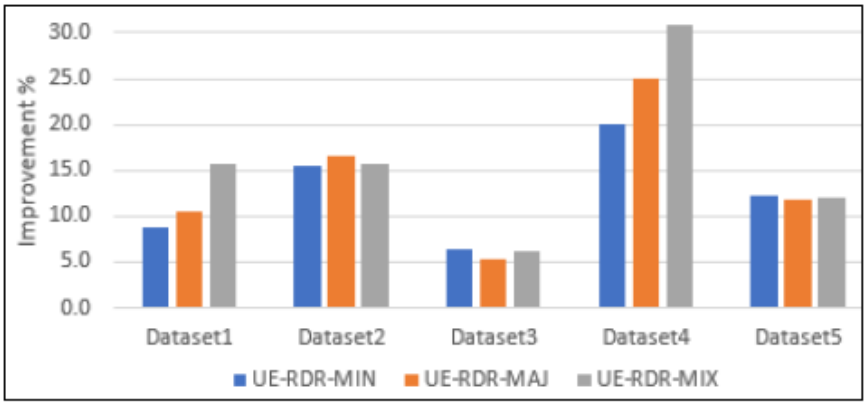


Figure 6: % Improvement in Ruleset Compactness over RIDOR.

The results show that compactness with all datasets is improved. However, UE-RDR-MIX compactness is better in Dataset 1 (Bank dataset) and Dataset 2 (Synthetic Bank dataset). For the remaining three datasets, either UE-RDR-MIN or UE-RDR-MAJ model performance is better.

IPA classifier accuracy for mixed Bank data is compared with UE-RDR-MIX model. Table 4 shows that UE-RDR accuracy is higher than IPA classifier.

Table 4: Accuracy comparison with IPA.

Technique	Accuracy
UE-RDR-MIX	83.76%
IPA(O. O. Maruatona, 2013)	73.90%

For further verification, the UE-RDR-MIX classification accuracy is also compared to a non-RDR classifier: Naïve Bayes. Figure 7 shows that UE-RDR accuracy is higher than Naïve Bayes accuracy for all datasets, with substantial improvements in accuracy for Datasets 1 and 4.

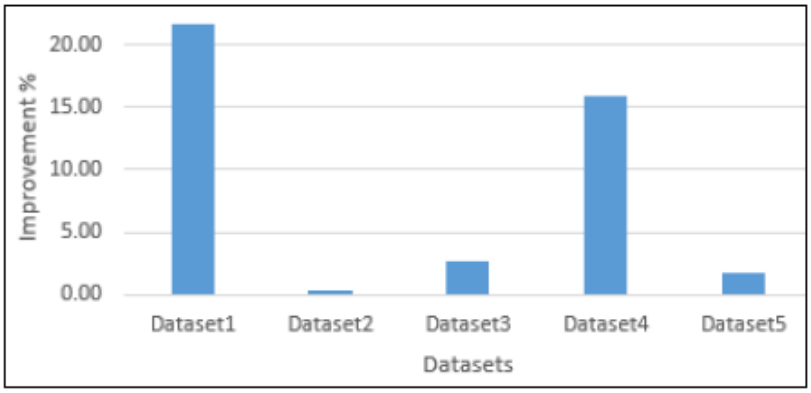


Figure 7: % Improvement in Classification Accuracy over Naïve Bayes.

Classification accuracy is compared among the three UE-RDR-models for a specific class label for mixed Bank data. Figure 8 shows that classification accuracy is higher with UE-RDR-MIX model.

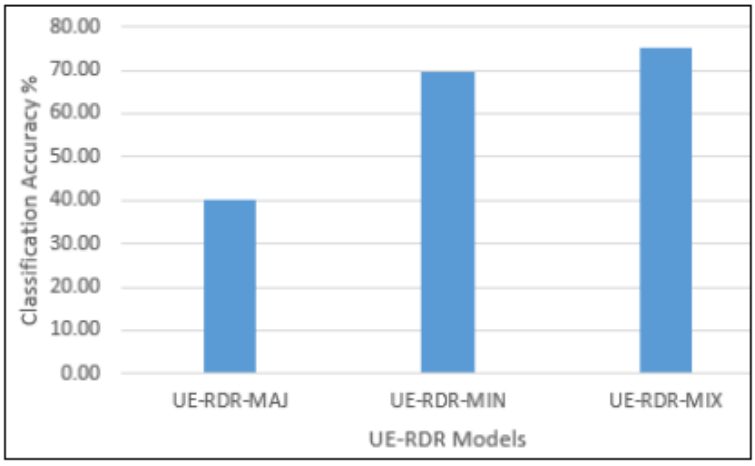


Figure 8: Classification accuracy in Fraud Class among UE-RDR models

Figure 5, Table 4 and Figure 6 shows that UE-RDR-MIX model gives best classification accuracy. While Figure 8 shows that a specific class label which is neither majority class nor minority class, also has a higher classification accuracy with UE-RDR-MIX model. The reason of higher accuracy is because of combined and compact rules in UE-RDR-MIX model for that class from majority and minority training models.

4 Conclusion

Fraud detection for online banking requires higher classification accuracy for the detection to enhance the confidence of its customers. Out of the available rule-based techniques for fraud detection, RDR is ideal due to its lower maintenance and incremental learning. However, testing and evaluating RDR on distributed and Big Data platform is a challenging task, as the RDR classifier has not yet been implemented on Spark. Paper has shown that, the challenge in fraud analysis due to the heterogeneous nature of transactions data (mixed attributes) and Big Data can be overcome with UE-RDR. Introducing Unified Expressions in the RDR and evaluating the expressions based on Lift score helped to achieve ruleset compactness and higher accuracy. Further three models, including UE-RDR-MIN, UE-RDR-MAJ and UE-RDR-MIX are also developed in this paper. UE-RDR-MIX is the most innovative model, which does not exist in RIDOR. It combines and further compacts Majority and Minority class models with least usage of default class and unlike RDR it contains rules of all class labels, so it gives better accuracy from RDR based classifiers.

Classification accuracy is compared with existing RDR implementation: RIDOR. This technique is applied on various datasets including fraud analysis Bank & Synthetic Bank datasets and three publicly available German Credit, Adult (Census Income) and Credit Approval datasets. The empirical evaluation has shown that not only the ruleset size of training and prediction dataset is reduced, but classification accuracy is also improved. Classification accuracy with UE-RDR for Bank dataset is also compared with another RDR based IPA technique and a non-RDR classifier (Naïve Bayes). Results have shown improvement in classification accuracy when compared with these classifiers as well. In this paper, the developed technique is used for the experimental validation and development of fraud analysis, but it can be used in other domains as well, especially for scalable and distributed systems. Further, this technique can be enhanced for other data formats (libsvm and arff) and a multi-classification system.

ACKNOWLEDGMENTS

This research was done in Internet Commerce Security Lab (ww.icsl.com.au), where Westpac bank, IBM, ACSC are partners.

REFERENCES

[1] ASF. (2015). Apache Hadoop. Retrieved from http://hadoop.apache.org/
[2] Compton, P. (2011). Pacific Knowledge Systems: Challenges with Rules. Retrieved from Sydney: http://pks.com.au/wp-content/uploads/2015/03/WhitePaperChallengesWithRulesPKS.pdf
[3] Compton, P., & Jansen, R. (1988). Knowledge in context: A strategy for expert system maintenance. Paper presented at the Australian Joint Conference on Artificial Intelligence, Adelaide, Australia.
[4] FBI. (2018). Internet Crime Complaint Center, 20182019(20/08/2019). Retrieved from https://pdf.ic3.gov/2018_IC3Report.pdf
[5] Gaines, B. R., & Compton, P. (1995). Induction of ripple-down rules applied to modeling large databases
[6] Journal of Intelligent Information Systems, 5(3), 211-228.
[7] Herland, M., Khoshgoftaar, T., & Bauder, R. (2018). Big Data fraud detection using multiple medicare data sources. Journal of Big Data, 5(1), 1-21. doi:10.1186/s40537-018-0138-3
[8] Hofmann, H. (1994). Statlog (German Credit Data) Data Set [Multivariate]. Financial. Retrieved from: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)
[9] Kang, B. H., Compton, P., & Preston, P. (1995, 1995). Multiple Classification Ripple Down Rules: Evaluation and Possibilities. Paper presented at the 9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, Banff.
[10] Kou, Y., Lu, C.-T., Sirwongwattana, S., & Huang, Y.-P. (2004). Survey of fraud detection techniques. Paper presented at the IEEE International Conference on Networking, Sensing and Control, 2004.
[11] Littin, J. N. (1996). Learning relational ripple-down rules. (PHD PHD), University of Waikato, Hamilton New Zealand. Retrieved from http://www.cs.waikato.ac.nz/~ml/publications/1996/JLittin96-Thesis.pdf
[12] Martinez, G. (2019). Lift (data mining). Retrieved from https://en.wikipedia.org/wiki/Lift_(data_mining)
[13] Maruatona, O., Vamplew, P., & Dazeley, R. (2012). RM and RDM, a Preliminary Evaluation of Two Prudent RDR Techniques. Paper presented at the Pacific Rim Knowledge Acquisition Workshop.
[14] Maruatona, O. O. (2013). Internet banking fraud detection using prudent analysis. (PHD PHD), University of Ballarat.
[15] McCombie, S. (2008). Trouble in Florida, The Genesis of Phishing attacks on Australian Banks. Paper presented at the 6th Australian Digital Forensics Conference., Perth.
[16] Melo-Acosta, G. E., Duitama-Munoz, F., & Arias-Londono, J. D. (2017). Fraud detection in big data using supervised and semi-supervised learning techniques. Paper presented at the 2017 IEEE Colombian Conference on Communications and Computing (COLCOM, Cartagena, Colombia.
[17] Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., . . . Owen, S. (2016). Mllib: Machine learning in apache spark. Journal of Machine Learning Research, 17(34), 1-7.
[18] Moloney, G., Barker, M., Coddington, P., & Mecoles, K. (2011). NECTAR. Retrieved from https://nectar.org.au
[19] Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. arXiv preprint arXiv:1009.6119.
[20] Prasad, Y. S., & Ramakrishna, G. (2016). A novel probabilistic based feature selection model for credit card anomaly detection. Journal of Theoretical and Applied Information Technology, 94(2), 335.
[21] Prayote, A. (2007). Knowledge Based Anomaly Detection. (PHD PHD), University of NSW.
[22] Quinlan, J. R. (1987). Simplifying decision trees. International journal of man-machine studies, 27(3), 221-234.
[23] Quinlan, J. R. (1992). C4.5: Programs for Machine Learning (Ist ed.): Morgan Kaufmann.
[24] Richards, D. (2003). Knowledge-based system explanation: The ripple-down rules alternative. Knowledge and Information Systems, 5(1), 2-25. doi:10.1007/s10115-002-0076-3
[25] Shanahan, J. G., & Dai, L. (2015, 2015). Large scale distributed data science using apache spark. Paper presented at the Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia.
[26] Swain, S. R., & Sarangi, S. S. (2013). Study of Various Classification Algorithms using Data Mining. International Journal of Advanced Research in Science and Technology (IJARST), 2(2), 110-114.

[27] Ul Haq, I., Gondal, I., & Vamplew, P. (2019). Enhancing Model Performance for Fraud Detection by Feature Engineering and Compact Unified Expressions. Paper presented at the 19th International Conference on Algorithms and Architectures for Parallel Processing, Melbourne, Australia.
[28] Ul Haq, I., Gondal, I., Vamplew, P., & Brown, S. (2018). Categorical Features Transformation with Compact One-hot Encoder for Fraud Detection in Distributed Environment. Paper presented at the The 16th Australasian Data Mining Conference, Bathurst NSW, Australia.
[29] Ul Haq, I., Gondal, I., Vamplew, P., & Layton, R. (2016). Generating Synthetic Datasets for Experimental Validation of Fraud Detection. Paper presented at the Fourteenth Australasian Data Mining Conference, Canberra, Australia.
[30] Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. Paper presented at the Proceedings of the twenty-first international conference on Machine learning.