Learning the sub-conceptual layer: A Framework for One-Class Classification

Shiven Sharma

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the Doctorate in Philosophy degree in Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

©Shiven Sharma, Ottawa, Canada, 2016

Abstract

In the realm of machine learning research and application, binary classification algorithms, *i.e.* algorithms that attempt to induce discriminant functions between two categories of data, reign supreme. Their fundamental property is the reliance on the availability of data from all known categories in order to induce functions that can offer acceptable levels of accuracy. Unfortunately, data from so-called "real-world" domains sometimes do not satisfy this property. In order to tackle this, researchers focus on methods such as sampling and cost-sensitive classification to make the data more conducive for binary classifiers.

However, as this thesis shall argue, there are scenarios in which even such explicit methods to rectify distributions fail. In such cases, one-class classification algorithms become a practical alternative. Unfortunately, if the domain is inherently complex, the advantage that they offer over binary classifiers becomes diminished. The work in this thesis addresses this issue, and builds a framework that allows for one-class algorithms to build efficient classifiers.

In particular, this thesis introduces the notion of learning along the lines sub-concepts in the domain; the complexity in domains arises due to the presence of sub-concepts, and by learning over them explicitly rather than on the entire domain as a whole, we can produce powerful one-class classification systems. The level of knowledge regarding these sub-concepts will naturally vary by domain, and thus we develop three distinct frameworks that take the amount of domain knowledge available into account. We demonstrate these frameworks over three real-world domains.

The first domain we consider is that of biometric authentication via a users swipe on a smartphone. We identify sub-concepts based on a users motion, and given that modern smartphones employ sensors that can identify motion, during learning as well as application, sub-concepts can be identified explicitly, and novel instances can be processed by the appropriate one-class classifier. The second domain is that of invasive isotope detection via gamma-ray spectra. The sub-concepts are based on environmental factors; however, the hardware employed cannot detect such con-

cepts, and quantifying the precise source that creates these sub-concepts is difficult to ascertain. To remedy this, we introduce a novel framework in which we employ a sub-concept detector by means of a multi-class classifier, which pre-processes novel instances in order to send them to the correct one-class classifier. The third domain is that of compliance verification of the Comprehensive Test Ban Treaty (CTBT) through Xenon isotope measurements. This domain presents the worst case where sub-concepts are not known. To this end, we employ a generic version of our framework in which we simply cluster the domain and build classifiers over each cluster. In all cases, we demonstrate that learning in the context of domain concepts greatly improves the performance of one-class classifiers.

Acknowledgment

Many individuals helped me along this most astonishing of journeys, and they deserve special acknowledgement. I wish to acknowledge the amazing support I received from my supervisors Dr. Nathalie Japkowicz and Dr. Anil Somayaji. Both have been instrumental for my research. Dr. Japkowicz introduced me to one-class learning and provided me with my first opportunity to work on a real-world problem. Middway during my thesis, when I was in need for finding a solid direction, Dr. Somayaji took me on and become a co-supervisor; my work with him was essential in narrowing down on a thesis topic. I would also like to thank Dr. Stan Matwin and Dr. John Oommen for providing me with valuable suggestions leading up to, and during my, thesis proposal.

The essence of this thesis was brone out from real-world projects, each of which presented challenges and insights that become key for developing my framework. I would like to acknowledge Health Canada and Zighra Inc for their assistance in this regards. Both organizations provided an exciting and fertile ground for developing and exploring my ideas, and were gracious to provide data and resources needed to complete my work. Finally, I would also like to thank Michael Bingham and Bheesham Persaud from the Carleton Computer Science Security Laboratory for developing the application for gathering swipes on iOS and Android platforms that was key for my work in one of the domains.

On a final note, I would like to dedicate this thesis to my late grandfather, Dr. Ram Prakash Sharma, and one of my best friends, Colin Bellinger. My grandfather supported me tremendously throughout my academic journey, providing me with both financial and moral support. He was very much looking forward to hearing the news when I would finally submit my thesis, and therefore I would like to dedicate this work to his memory. Colin Bellinger has been a brother me, both in my professional and personal life, and for that, I am eternally grateful.

Contents

ABSTRACT ii
ACKNOWLEDGMENT iv
LIST OF TABLES x
LIST OF FIGURES xviii

1 Introduction **1**
1.1 Motivation: Real-world problems 3
1.1.1 Extreme Imbalance 4
1.1.2 Complex Distributions 5
1.2 Thesis Contribution 6
1.3 Statement of collaborative work and publications 8
1.4 Organisation 9

2 Learning in Imbalanced Domains **10**
2.1 The Imbalance Problem 10
2.2 Imbalanced Learning with Multiple Classes 12
2.2.1 Sampling Methods 13
2.2.2 Kernel Based Methods 15
2.2.3 Cost-Sensitive Methods 15
2.3 Imbalanced Learning with One-Class Classification 16
2.3.1 An Introduction to One-Class Classification 17
2.3.2 Other Methods for Outlier Detection 22
2.4 Challenges in Imbalanced Domains 25
2.4.1 Imbalance and Overlap 26

2.4.2 Imbalance and Multi-Modality: Divide-And-Conquer 27
2.5 Summary 32

3 One-Class versus Binary Classification **34**
3.1 One-Class versus Binary Classification: A Bayesian Perspective . . . 35
3.2 Description of the Data Sets 36
3.2.1 Artificial Data 36
3.2.2 UCI Datasets 37
3.3 Experimental Framework 48
3.4 Results 50
3.4.1 Results on *Artificial Unimodal* dataset 50
3.4.2 Results on *Artificial Multimodal* datasets 52
3.4.3 Results on *Diabetes* dataset 54
3.4.4 Results on *Heart Disease* dataset 56
3.4.5 Results on *Ionosphere* dataset 57
3.4.6 Results on *Thyroid Disease* dataset 57
3.4.7 Results on *Sonar* dataset 60
3.4.8 Results on *Alphabets* dataset 61
3.4.9 Results on *Forest* dataset 63
3.4.10 Results on *ForestC1* dataset 64
3.4.11 Results on *ForestC2C5* dataset 66
3.5 Discussion 67
3.6 Summary 69

4 Learning the Sub-Conceptual Layer **70**
4.1 One-Class Classification over Sub-Concepts 71
4.1.1 One-Class Classification with Complete Knowledge 75
4.1.2 One-Class Classification with Fuzzy Knowledge 77
4.1.3 One-Class Classification with No Knowledge 79
4.1.4 One-Class Classification along Sub-Concepts: A Mathematical Perspective 81

4.2 Experimental Framework 83
4.3 Results 85
4.3.1 Results on *Artificial Multimodal* datasets 86
4.3.2 Results on *Diabetes* dataset 87
4.3.3 Results on *Heart Disease* dataset 88
4.3.4 Results on *Ionosphere* dataset 89
4.3.5 Results on *Thyroid Disease* dataset 89
4.3.6 Results on *Sonar* dataset 91
4.3.7 Results on *Alphabets* dataset 92
4.3.8 Results on *Forest* dataset 93
4.3.9 Results on *ForestC1* dataset 94
4.3.10 Results on *ForestC2C5* dataset 94
4.3.11 Statistical Analysis 95
4.4 Discussion 96
4.5 Summary 100

5 Biometrics for Security **101**

5.1 Biometrics: An Overview 102
5.1.1 Behavioural Biometrics in Smartphones 104
5.2 Swipes as a behavioural biometric for mobile phones 106
5.2.1 Swipe Representation 108
5.3 The Nuances of Swipes 111
5.3.1 Learning Complexity 111
5.3.2 Domain Complexity 112
5.3.3 Other Considerations 116
5.4 Learning Framework 117
5.5 Experimental Framework 119
5.6 Results 120
5.6.1 Discussion 121
5.7 Conclusions 123

6 Case study - gamma-ray spectra data **124**

6.1 Gamma-Ray Spectrometer Data . . . 125

6.1.1 Learning Complexity . . . 125

6.1.2 Domain Complexity . . . 127

6.1.3 Summary . . . 131

6.2 Learning Framework . . . 131

6.2.1 Phase I: Rain separation . . . 131

6.2.2 Phase II: Isotope/Anomaly Detection . . . 133

6.2.3 Threshold Selection . . . 134

6.3 Experimental Framework . . . 134

6.3.1 Data Pre-processing . . . 135

6.3.2 Training and Testing Procedures . . . 135

6.4 Results . . . 136

6.4.1 Results from the Two-Tier Anomaly Detection System . . . 137

6.4.2 Results from no rain separation . . . 140

6.4.3 Discussion . . . 140

6.5 Conclusions . . . 142

7 Case Study - The CTBT Domain **144**

7.1 Domain Description . . . 145

7.1.1 Learning Complexity . . . 145

7.1.2 Domain Complexity . . . 147

7.1.3 Summary . . . 147

7.2 Experimental Framework . . . 148

7.3 Results . . . 148

7.4 Conclusions . . . 149

8 Concluding Remarks **151**

8.1 Simplification with domain knowledge . . . 152

8.2 Simplification without domain knowledge . . . 153

8.3 Revisiting the thesis contribution . . . 154

8.4 Future Work 155
8.4.1 Feature Selection for sub-concepts 155
8.4.2 Combining frameworks 155
8.4.3 Classifier ensembles in the absence of knowledge 156
8.4.4 One-class classifiers as substitutes for binary classifiers 157
8.4.5 Learning with multiple classes with Imbalance 157
8.5 A Final Note 158

A Full opacity plots for the Alphabets, Forest, ForestC1 and ForestC2C5 datasets **159**

B Sampling plots for binary classifiers **162**
B.1 Results on *Artificial Unimodal* dataset 162
B.2 Results on *Artificial Multimodal* dataset with no overlap 162
B.3 Results on *Artificial Multimodal* dataset with overlap 166
B.4 Results on *Diabetes* dataset 168
B.5 Results on *Heart Disease* dataset 170
B.6 Results on *Ionosphere* dataset 172
B.7 Results on *Sonar* dataset 174
B.8 Results on *Thyroid Disease* dataset 176
B.9 Results on *Alphabets* dataset 178
B.10 Results on *Forest* dataset 180
B.11 Results on *ForestC1* dataset 182
B.12 Results on *ForestC2C5* dataset 184
BIBLIOGRAPHY 195

List of Tables

3.1 Description of the UCI datasets. 39

4.1 Frameworks employed for each dataset 85
4.2 Results of the Wilcoxon Signed Ranks test for the clustered and regular versions of the one-class classifiers. 96
4.3 Results of the Wilcoxon Signed Ranks test for the fuzzy knowledge and regular versions of the one-class classifiers. 96
4.4 Summary of results over all domains 97
4.5 The target and outlier class accuracies over the various domains . . . 99

5.1 g-mean values for different users 116
5.2 Authentication accuracies for different motions for User 1. 121
5.3 Authentication accuracies for different motions for User 2. 121

6.1 AUC values for all Stations, over all binary classifiers, for both Rain and Non-Rain anomaly detection systems. 138
6.2 AUC Values from no rain separation. 140
6.3 Total number of True Positives (TPs) and False Positives (FPs) incurred with (2-Tier) and without (1-Tier) rain separation. 141

List of Figures

2.1 Architecture of an Autoassociator 19
2.2 Density and Class Probability Estimation 21
2.3 The One-Class Support Vector Machine 22
2.4 Illustration of the Mahalanobis Distance between two points **A** and **B** from the mean. Both points have the same Mahalanobis distance, but different Euclidean distances from the mean. 23
2.5 An overview of research coverage in the field of learning over imbalanced domains. 33

3.1 The first three principal components of the *unimodal artificial* dataset. The target class exists as a unimodal gaussian, whereas the outlier class has four modes. 38
3.2 The first three principal components of the *multimodal artificial* dataset with no overlap. Both the target and outlier classes exist as distributions with 4 modes with minimal overlap between each class. 38
3.3 The first three principal components of the *multimodal artificial* dataset with overlap. Both the target and outlier classes exist as distributions with 4 modes with significant overlap between each class. 39
3.4 The first three principal components of the *Diabetes* dataset. 40
3.5 The first three principal components of the *Heart Disease* dataset. . . 41
3.6 The first three principal components of the *Ionosphere* dataset. 42
3.7 The first three principal components of the *Thyroid Disease* dataset. . 43
3.8 The first three principal components of the *Sonar* dataset. 43

3.9 The first three principal components of the *Alphabets* dataset. 44
3.10 The first three principal components of the *Forest* dataset. 46
3.11 The first three principal components of the *ForestC1* dataset. 47
3.12 The first three principal components of the *ForestC2C5* dataset. . . . 47
3.13 The performance trends of various classifiers over the artificial unimodal dataset with no sampling. 51
3.14 The performance trends of various classifiers over the artificial unimodal dataset with undersampling. 51
3.15 The performance trends of various classifiers over the artificial multimodal dataset (no overlap) with no sampling. 52
3.16 The performance trends of various classifiers over the artificial multimoda dataset (no overlap) with SMOTE. 53
3.17 The performance trends of various classifiers over the artificial multimodal dataset (with overlap) with no sampling. 53
3.18 The performance trends of various classifiers over the artificial multimoda dataset (with overlap) with SMOTE. 54
3.19 The performance trends of various classifiers over the diabetes dataset with no sampling. 55
3.20 The performance trends of various classifiers over the diabetes dataset with undersampling. 55
3.21 The performance trends of various classifiers over the heart disease dataset with no sampling. 56
3.22 The performance trends of various classifiers over the heart disease dataset with undersampling. 57
3.23 The performance trends of various classifiers over the ionosphere dataset with no sampling. 58
3.24 The performance trends of various classifiers over the ionosphere dataset with undersampling. 58
3.25 The performance trends of various classifiers over the thyroid disease dataset with no sampling. 59

3.26 The performance trends of various classifiers over the thyroid disease dataset with undersampling. 59
3.27 The performance trends of various classifiers over the sonar dataset with no sampling. 60
3.28 The performance trends of various classifiers over the sonar dataset with undersampling. 61
3.29 The performance trends of various classifiers over the alphabets dataset with no sampling. 62
3.30 The performance trends of various classifiers over the alphabets dataset with undersampling. 62
3.31 The performance trends of various classifiers over the forest dataset with no sampling. 63
3.32 The performance trends of various classifiers over the forest dataset with undersampling. 64
3.33 The performance trends of various classifiers over the forestC1 dataset with no sampling. 65
3.34 The performance trends of various classifiers over the forestC1 dataset with undersampling. 65
3.35 The performance trends of various classifiers over the forestC2C5 dataset with no sampling. 66
3.36 The performance trends of various classifiers over the forestC2C5 dataset with undersampling. 67

4.1 The notion of main and sub-concepts. 72
4.2 Inducing a classifier without distinguishing between different sub-concepts. 74
4.3 Inducing classifiers by distinguishing between different sub-concepts. . 74
4.4 One-class classification with full knowledge 76
4.5 One-class classification with fuzzy knowledge. 78
4.6 One-class classification with no knowledge. 80

4.7 The performance values of various framework over the artificial multimodal dataset with no overlap. 86
4.8 The performance values of various framework over the artificial multimodal dataset with overlap. 87
4.9 The performance values of various classifiers over the diabetes dataset. 88
4.10 The performance values of various classifiers over the heart disease dataset. 89
4.11 The performance values of various classifiers over the ionosphere dataset. 90
4.12 The performance values of various classifiers over the thyroid disease dataset. 90
4.13 The performance values of various classifiers over the sonar dataset. . 91
4.14 The performance values of various classifiers over the alphabets dataset. 92
4.15 The performance values of various classifiers over the forest dataset. . 93
4.16 The performance values of various classifiers over the forestC1 dataset. 94
4.17 The performance values of various classifiers over the forestC2C5 dataset. 95

5.1 Lock screens that employ sub-conscious swipes for the Samsung Galaxy Note in (*i*) and the iPhone in (*ii*). 107
5.2 x-axis accelerometer data for three users. 108
5.3 Touch Features 109
5.4 Discretization of the accelerometer and gyroscope time series. 110
5.5 The first three principal components for the accelerometer features for Users 1 and 2. 113
5.6 The first three principal components for the gyroscope features for Users 1 and 2. 113
5.7 The first three principal components for the accelerometer features for User 1. 114
5.8 The first three principal components for the gyroscope features for User 1. 114

5.9 The first three principal components for the accelerometer features for User 2. 115
5.10 The first three principal components for the gyroscope features for User 2. 115
5.11 Accelerometer time series for walking and sitting. 122
5.12 Gyroscope time series for sitting and walking. 122

6.1 These figures contain randomly selected examples from the gamma-ray spectra data set. Plotted on the log-scale, sub-figure *(i)* depicts a background instance and sub-figure *(ii)* depicts and instance containing the medical isotope Technetium. 126
6.2 Mean rain and non-rain spectra for Station 6. Blue represents rain and red non-rain. 128
6.3 Mean rain and non-rain spectra for Station 12. Blue represents rain and red non-rain. 128
6.4 Mean rain and non-rain spectra for Station 13. Blue represents rain and red non-rain. 129
6.5 The first three principal components for Station 6. Red represents no rain and blue represents rain. 129
6.6 The first three principal components for Station 12. Red represents no rain and blue represents rain. 130
6.7 The first three principal components for Station 13. Red represents no rain and blue represents rain. 130
6.8 The proposed architecture for anomaly detection in gamma-rays. . . . 132
6.9 This illustrates the complete experimental framework undertaken to validate our proposed architecture. 137
6.10 The ROC plot for Station 13, generated using the results obtained by utilizing the techniques employed at Health Canada. 138
6.11 The ROC plot for Station 6 and Station 12, generated using the results obtained by utilizing the techniques employed at Health Canada. . . . 138

6.12 The ROC curves for Station 6 for both rain and non-rain events, using the Naïve Bayes Classifier and the Mahalanobis Distance.. 139
6.13 The ROC curves for Station 12 for both rain and non-rain events, using the Naïve Bayes Classifier and the Mahalanobis Distance. 139
6.14 The ROC curves for Station 13 for both rain and non-rain events, using the Naïve Bayes Classifier and the Mahalanobis Distance.. 140

7.1 The first three principal components of the CTBT data. 147
7.2 Results of the clustered and non-clustered (normal) autoassociator and PDEN over the CTBT dataset. 149

A.1 The first three principle components of the *Alphabets* dataset. 159
A.2 The first three principle components of the *Forest* dataset. 160
A.3 The first three principle components of the *ForestC1* dataset. 160
A.4 The first three principle components of the *ForestC2C5* dataset. . . . 161

B.1 The performance trends of various classifiers over the artificial unimodal dataset with oversampling. 163
B.2 The performance trends of various classifiers over the artificial unimodal dataset with one-sided selection. 163
B.3 The performance trends of various classifiers over the artificial unimodal dataset with SMOTE. 164
B.4 The performance trends of various classifiers over the artificial multimodal dataset with oversampling. 164
B.5 The performance trends of various classifiers over the artificial multimodal dataset with one-sided selection. 165
B.6 The performance trends of various classifiers over the artificial multimodal dataset with undersampling. 165
B.7 The performance trends of various classifiers over the artificial multimodal dataset with oversampling. 166

B.8 The performance trends of various classifiers over the artificial multimodal dataset with one-sided selection. 167
B.9 The performance trends of various classifiers over the artificial multimodal dataset with undersampling. 167
B.10 The performance trends of various classifiers over the diabetes dataset with oversampling. 168
B.11 The performance trends of various classifiers over the diabetes dataset with one-sided selection. 169
B.12 The performance trends of various classifiers over the diabetes dataset with SMOTE. 169
B.13 The performance trends of various classifiers over the heart disease dataset with oversampling. 170
B.14 The performance trends of various classifiers over the heart disease dataset with one-sided selection. 171
B.15 The performance trends of various classifiers over the heart disease dataset with SMOTE. 171
B.16 The performance trends of various classifiers over the ionosphere dataset with oversampling. 172
B.17 The performance trends of various classifiers over the ionosphere dataset with one-sided selection. 173
B.18 The performance trends of various classifiers over the ionosphere dataset with SMOTE. 173
B.19 The performance trends of various classifiers over the sonar dataset with oversampling. 174
B.20 The performance trends of various classifiers over the sonar dataset with one-sided selection. 175
B.21 The performance trends of various classifiers over the sonar dataset with SMOTE. 175
B.22 The performance trends of various classifiers over the thyroid disease dataset with oversampling. 176

B.23 The performance trends of various classifiers over the thyroid disease dataset with one-sided selection. 177
B.24 The performance trends of various classifiers over the thyroid disease dataset with SMOTE. 177
B.25 The performance trends of various classifiers over the alphabets dataset with oversampling. 178
B.26 The performance trends of various classifiers over the alphabets dataset with one-sided selection. 179
B.27 The performance trends of various classifiers over the alphabets dataset with SMOTE. 179
B.28 The performance trends of various classifiers over the forest dataset with oversampling. 180
B.29 The performance trends of various classifiers over the forest dataset with one-sided selection. 181
B.30 The performance trends of various classifiers over the forest dataset with SMOTE. 181
B.31 The performance trends of various classifiers over the forestC1 dataset with oversampling. 182
B.32 The performance trends of various classifiers over the forestC1 dataset with one-sided selection. 183
B.33 The performance trends of various classifiers over the forestC1 dataset with SMOTE. 183
B.34 The performance trends of various classifiers over the forestC2C5 dataset with oversampling. 184
B.35 The performance trends of various classifiers over the forestC2C5 dataset with one-sided selection. 185
B.36 The performance trends of various classifiers over the forestC2C5 dataset with SMOTE. 185

Chapter 1

Introduction

With the proliferation of data collection over the past few decades, the ubiquity of data in most domains provides an exceptionally conducive platform for practical research into the application of machine learning algorithms, particularly classification, *i.e.*, automatically classifying novel datum into the class/category to which it belongs. Industry is increasingly interested in employing intelligent algorithms for fast and efficient solutions to tasks that would typically require vast human resources or cumbersome and outdated approaches. As a result, machine learning research is now being exposed to the intricacies and nuances that large amounts of data from real-word domains presents.

However, with this exposure, many of the standard, implicit assumptions that machine learning methods often make are now being torn apart; real-world data does not play by these assumed rules. This has a direct consequence on how algorithms and ideas from machine learning can be applied to these real-world domains. Specifically, algorithms need to be tuned and developed that not only handle the issues associated with real-world data, but are also able to take advantage of the nuances inherent in it in order to maximize performance.

The challenges that real-world data brings, as alluded to in the previous paragraph, can arise from various aspects, including the noise in the data, complexity of the data with respect to its underlying probability distribution function, class imbalance, dimensionality, etc. In this thesis, we consider two major aspects that can

have a severe negative impact on machine learning solutions. The first, and perhaps one of the most significant and pervasive issue, is that of *class imbalance*. Multi-class classification algorithms that form the core of the machine learning world assume relatively balanced distributions of categories (classes) in order to induce appropriate discriminant functions. Put more simply, there is sufficient data from all known categories that make up the domain to build a model that can classify novel data with acceptable accuracy. But, more often than not, real-world data is far from balanced; there is, typically, an over abundance of data from certain classes, and very little data from other classes. The issues arising directly, or indirectly, from such imbalanced distributions deserve special attention and, of late, research into learning in such domains has gained in popularity.

In this thesis we examine the impact of of imbalance over domains that conform to a binary classification formulation, *i.e.*, there are only two classes in the domain. Binary classification is the simplest form of multi-class classification; indeed, most of the theoretical work in classification is conducted under the umbrella of binary classification. Furthermore, many real-world domains, especially those from the world of security, naturally conform to a binary class formulation. Binary classification, thus, is a logical starting point for studying the impacts of imbalance on learning algorithms.

In order to rectify imbalance, methods employed typically include oversampling, undersampling and cost-sensitive classification. For these methods to be applicable, it is assumed that there is data available from both classes, albeit with an overabundance of data from one class. However, there are cases when the only data available to learn is from a single class. This can be either due to an inability to procure data in a practical manner from the second class, or due to a severe dearth in the data that is available. In such scenarios, one-class classification, a special paradigm of classification algorithms, becomes the only viable framework for learning. These classifiers aim to generalize over the data available from the known class (typically referred to as the target class), thus performing classification by recognition; if the model is able to recognize a novel datum, it is assigned to the target class, otherwise it is classified

as belonging to the unknown class (typically referred to as the outlier class). However, given that one-class classifiers attempt to generalize over the data space of a single class, the more complex the space, the harder the task of generalization will be. This one-class classifiers are impacted by the second issue that we consider, that of *domain complexity*. Overlap between the various classes and multi-modality can cause the domain to have a highly complex distribution. This has a significant impact on one-class classifiers as the more complex the data distribution, the harder it is to model it.

These two observations underlie the work presented in this thesis. Our work was inspired by research conducted in three real-world domains, all of which suffered from the aforementioned issues. Section 1.1 presents the observations that motivated our work into improving the performance of the learning framework in these domains. Section 1.2 outlines the work presented in the thesis and our contribution to the area of real-world machine learning. Finally, the organization of the thesis is outlined in Section 1.4

1.1 Motivation: Real-world problems

During the course of our research, we explored three domains that exemplified the aforementioned observations. The first domain related to biometric security on mobile phones, and used biometric sensor data gathered from a mobile phone's accelerometer, gyroscope and touch screen during a swipe action, the aim being to employ this to authenticate only the user of the phone. The second domain comprised of gamma-ray spectra, and involved investigating the applicability of machine learning for the purposes of detecting gamma-ray signatures emitted from dangerous isotopes; for example, Uranium or Plutonium. The third domain comprised of data representing Xenon isotopes, and dealt with the compliance verification of the Comprehensive Test Ban Treaty (CTBT) [75, 76]; we were tasked with investigating whether machine learning could be applied in order to automate the detection of clandestine nuclear tests by nations. In the following, we will highlight the major research chal-

lenges (specifically, those detailed at the start of the chapter) that, in turn, became motivations for the research undertaken in this thesis.

1.1.1 Extreme Imbalance

All domains considered conform to a binary classification formulation; there is a *target* class, which is the class for which we have available data, and there is an *outlier* class, the class for which we do not have data. In the domain of mobile security, only the owner of the phone will generate swipe data; data from users other than the phone's owner is non-existent. A similar situation occurred in the second research project that dealt with identifying gamma-ray signatures of malicious isotopes, which also had two classes. As one would expect, the likelihood of individuals walking with dangerous isotopes is extremely low, and as a result, data from this set was also exceptionally scarce. Finally, in the domain of CTBT verification, there were two classes: a *background* class pertaining to normal radiation, and an *alarm* class indicating evidence of clandestine nuclear testing. Naturally, given the modern political climate, instances of the latter class would be few and far between, since extremely few nations conduct nuclear tests openly. As a result, the levels of imbalance were extreme. There was not sufficient data of the alarm class to even consider applying standard imbalance rectifying techniques.

These domains represent an extreme form of imbalance, one in which data from classes of "interest" are either impossible to collect (or even if instances occur, there are very few of them available to be of any use for inducing classifiers), or represent a set of an infinite cardinality. The CTBT and gamma-ray signature domains display the former form of imbalance. The latter form occurs in domains such as password hardening or typist verification [32] or network intrusion detection [34], and the domain of mobile security. In these domains, the attacker class can have any number of different signatures. In order to use techniques for dealing with imbalance, such as under- or over-sampling, there has be at least *some* amount of data present. Unfortunately, this is not the case in such domains. Therefore, one-class classification,

or anomaly detection, becomes the only viable option[1]; we attempt to model the majority class, which, in essence, is the only class available.

1.1.2 Complex Distributions

During our analysis of the sensor data recorded during a users swipe gesture on their phone, we observed that their motion had a significant impact on the nature of the data; swipe readings were significantly different during walking as opposed to during sitting or standing. Thus, the overall distribution of swipe readings gathered over time became inherently complex. In the gamma-ray domain, the complexity of the background class stemmed from weather. Specifically, rain had a significant impact, and as a result, the background class formed a bimodal distribution. Finally, when we were conducting experiments to assess the performance of one-class classifiers on the CTBT domain, we observed that the classifiers were unable to model the distributions as accurately as we would hope. We hypothesised that this was due to the background class having a very complex distribution, which the one-class classifiers were unable to generalize efficiently.

Discriminatory classifiers, by their very nature, are able to handle complex distributions relatively well. For example, neural networks can learn highly non-linear functions in order to separate different classes. One-class classifiers, on the other hand, fare poorly, as they attempt to learn one model for what is potentially a distribution comprised of multiple sub-concepts [70]. Thus, even though we may have narrowed in on the right paradigm for learning, *one-class classification*, there still remains the challenge of employing that paradigm over complex distributions.

It is important to stress that the complexity arising due to different concepts in the data over which we build one-class classifiers pertains to the entire domain, and thus, affects all classes, targets as well as outliers; the concepts that we refer to throughout this thesis are defined over the domain. Naturally, given that we only have data available from a single class, only that particular class can be split along the

[1]In this thesis, we will use the phrases *one-class classification*, *anomaly detection* and *outlier detection* interchangeably.

lines of these concepts. However, it is prudent to take into consideration the concept to which outlier instances belong as well. Consider, for instance, the swipe-based authentication domain. As we will discuss in subsequent chapters, walking tends to create a highly variable distribution, whereas swipes done while stationary conform to a very "tight" distribution. Consequently, thresholds selected will reflect this aspect of the distribution. If an attacker was to swipe while stationary, and the model employed to authenticate is the one learnt while walking, there is a high likelihood of the attacker being authenticated, as the threshold for a variable distribution will naturally be higher than that for a tighter distribution. Thus, it is prudent to consider an attacker's swipe under the concept that underlies it; if we were to authenticate a stationary attacker with the users stationary model, the likelihood of the attacker getting accept is greatly reduced (as we will demonstrate in later chapters).

1.2 Thesis Contribution

We have highlighted the nuances inherent in many real-world datasets that preclude the application of a binary classification algorithms. This is primarily due to the lack of data available to induce discriminative classifiers. The bulk of the work in dealing with imbalanced datasets has been focused on assuming that there is *enough* data from the minority classes; under such situations the distributions can be modified to take advantage of binary classification algorithms [31]. As a result, research into understanding the nature of one-class classification has never garnered much interest, the premise being that datasets can be modified to make them conducive for binary classification. Furthermore, research studies typically compare one-class classifiers to binary classifiers, as in [33], [40] and [58]. Such studies, in our view, do not hold much meaning as the comparisons are being conducted using datasets which can indeed be used by binary classifiers, or whose imbalance can be rectified using methods discussed in Chapter 2. As we have stressed, there are situations when such modifications are impossible to apply as there is only sufficient data from a single class. One-class classification is a learning paradigm typically employed in such scenarios, and in

Chapter 3, we analyze in detail the efficacy of employing this over binary classifiers as the imbalances levels become extreme.

However, establishing the situations under which one-class classification is the suitable choice is one matter; actually employing it is an entirely different matter altogether. Modelling a single class that exhibits a complex distribution is a typically hard task, as was noted by the poor performance of one-class classifiers over the domains described previously. In other words, the advantage that one-class classifiers possess over binary classifiers in domains of imbalance disappears when the domain in consideration is complex. Specifically, both multi-modality and class overlap can cause one-class classifiers to suffer in learning appropriate models. We propose that these complexities can be attributed to the presence of implicit concepts (sub-concepts) within the domain, and can be resolved by employing domain-specific heuristics and categories, particularly with the help of a domain expert. The advantage of possessing domain-specific knowledge is that we are able to have *a priori* knowledge as to the sub-concepts expected within the domain. In the domain of mobile security, the division is based on whether a user is in motion or stationary. Mobile devices are able to detect motion, and thus we can employ this ability to facilitate this division on the device. Over the gamma-ray spectra domain, we utilize environmental knowledge to divide the domain into two sub-concepts. Specifically, we divide based on readings impacted by the presence of water, and all other readings.

Thus, the core idea is to simplify the original domain into sub-domains that are easier to model; learning in the context of concepts rather than over the entire domain will narrow the focus of the learner, allowing it to be more effective at accepting target class instances and rejecting data from all other classes. The most general approach, one which does not require any domain-specific knowledge, is to cluster the original space and induce models over each cluster. We formalize the notion of concepts and how to utilize them to improve one-class classifiers in Chapter 4. In particular, we provide a mathematical formulation for the most general hypothesis and illustrate the various framework on artificial and UCI datasets.

To conclude, we can summarize our thesis contributions as:

- We empirically analyze the situations under which one-class classification is the only viable option (Chapter 3).

- We develop three frameworks, based on the extent of available domain knowledge regarding the presence of sub-concepts, that improve the performance of one-class classifiers (Chapter 4).

- In order to demonstrate their applicability in practice, we apply the aforementioned frameworks over three real-world domains (Chapters 5, 6 and 7).

1.3 Statement of collaborative work and publications

The research conducted over the real-world domains was not done in isolation; effective research is collaborative research, and the work in this thesis is no different. This section lists the collaborations entered into along with any associated publications.

Chapter 4 The genesis of the framework was published in our work in Sharma *et. al* [70]; the main domain over which the research was done was introduced as a machine learning competition by Health Canada at ICDM 2008 [75]. The published work won the *Best Paper Award* at the 25^{th} Canadian Conference on Artificial Intelligence.

Chapter 5 This chapter is related to biometric security on mobile phones. The notion of employing swipes as a biometric for mobile security started as a joint project between Zighra Inc.[2] and the Carleton Computer Security Laboratory[3]. A key challenge from a usability perspective was generating an appropriate interface for recording swipe on the phone, and this was addressed and researched in Bingham [10] (*thesis in defense*).

[2] `http://www.zighra.com/`
[3] `https://www.ccsl.carleton.ca/about/`

Chapter 6 The work in this chapter was a collaborative work with Health Canada, in particular, with Dr. Kurt Ungar, Mr. Rodney Berg and Colin Bellinger. Our preliminary work on employing machine learning solutions for the project was published in Sharma *et. al* [71].

1.4 Organisation

Chapter 2 provides a brief introduction to the problem of learning in imbalanced domains, including the problem of *small disjuncts*, and various methods that have been employed to tackle the problem, such as sampling and one-class classification. The applicability of these methods (specifically, sampling) in domains of extreme imbalance is analyzed in Chapter 3. Chapter 4 provides a mathematical and empirical analysis for the general framework for improving one-class classifier performance by dividing the domain along sub-concepts. The empirical analysis is conducted over artificial and UCI data sets.

We then present three case studies over real-world domains that exhibit the properties elaborated in Chapter 3. We first apply our framework for the problem of swipe-based authentication; this is discussed in Chapter 5. In Chapter 6 we consider the domain of invasive isotope detection through gamma-ray spectra. Finally, in Chapter 7, we consider our framework within the general context when no domain-specific knowledge is available. In particular, we employ clustering to simplify the CTBT domain. Chapter 8 provides concluding remarks and possible direction for future research.

Chapter 2

Learning in Imbalanced Domains

This chapter presents an overview of research done in the area of classification in domains that possess imbalanced data distributions. The research can be considered to be focused along two distinct lines: learning with two classes (discriminatory algorithms), and learning with only a single class (recognition algorithms). The former is concerned with rectifying the imbalance such that the domain becomes conducive for binary classifiers, whereas the latter aims to discover and improve algorithms that are able to learn with data from a single (available) distribution.

We begin this by describing the problem of imbalance and highlighting the different types of imbalance, elaborating on the concept of *small disjuncts*. Then, we proceed to discuss the various strategies employed thus far for dealing with imbalance. Once the foundations for theory are established we consider the various challenges encountered in imbalanced domains. Specifically, we look at research examining the nuances of learning in complex domains and understanding the relationship between overlap with imbalance. We conclude this chapter with a summary of the related work, explicitly stating the gaps that exist which the work presented in this thesis aims to fill.

2.1 The Imbalance Problem

The *class-imbalance problem* [39] is a pervasive issue in Machine Learning. A survey of the literature us with two different facets of the imbalance in datasets. The first

type of imbalance is termed *between-class imbalance*, and exhibits itself when the size of one class is considerably larger than the size of the other class. This type of imbalance is perhaps more explicitly obvious, as it manifests itself by skewing the distribution of instances between the classes. For example, a dataset might have $1,000$ instances of Class A and 10 instances of Class B. Examples of domains in which we find a between-class imbalance include oil-spill detection, network intrusion detection, fraud detection and medicine (see [51], [65], [13], [34]). Between-class imbalance is also referred to as imbalance due to *rare classes* [84].

The second type of imbalance noted in literature is one which occurs due to *rare cases*, and is referred to as *within-class* imbalance. Rare cases correspond to regions in the instance space that are inherently sparse; very few instances occur in these regions as compared to other regions, and they represent sub concepts within larger concepts. If one were to view a single class as being comprised of multiple sub-classes, then this form of imbalance can be viewed as a *between-class* imbalance between the sub-classes. The existence of rare cases gives rise to the problem of *small disjuncts*. Classifiers learn concept definitions by creating a number of disjunctive rules that cover a vast majority of examples that pertain to the concept. However, if there are sparse regions in the instance space, rules are created that cover a tiny cluster of examples, *i.e.* small disjuncts. The *problem with small disjuncts* is that they tend to have a much higher error rate than large disjuncts [84]. It should be noted that the problem with small disjuncts is not limited to the minority class; in fact, it is more of an issue in the majority class as, given the much larger number of instances present in it, the likelihood of there being sparse regions within the instance space is considerably higher.

It is important to make a distinction between *relative rarity* and *absolute rarity*. Consider an example in which Class A has $1,000,000$ instances, and Class B has $1,000$ instances. Clearly, there is an imbalance between the classes, a ratio of $1 : 1000$. However, note that the severity resulting from this imbalance would not be that detrimental for classifier induction as one might imagine; there are still $1,000$ instances from Class B, which would be, in most cases, sufficient enough to induce a classifier.

Therefore, the rarity is *relative* to the majority class. Consider now another example in which Class A has $1,000$ instances and Class B has 1. Once again, there is an imbalance, but note that the severity resulting from it is far more detrimental for concept induction as compared to the previous example; only a single instance is available from Class B! This represents an *absolute rarity*.

Depending on the level of imbalance, the induced classifier will be biased towards the majority class [50, 82]. To see why, consider the 1-nearest neighbour classifier. This classifier will assign to a novel instance the class label of its nearest neighbour. In a heavily imbalanced scenario, the likelihood that an instance belonging to the minority class will have a majority class instance as its nearest neighbour will be high[1]. Consequently, the classifier will misclassify a large number of minority class instances. Thus, dealing with imbalance is of paramount importance in order for efficient classifiers to be induced. In subsequent sections, the various methods employed to handle imbalance will be outlined. We begin by highlighting methods that employ all classes in the domain followed by one-class classifiers.

2.2 Imbalanced Learning with Multiple Classes

The power of multi-class classifiers is well established in the machine learning community. However, as we discussed in the previous section, this power diminishes in the face of imbalance. In this section, we provide an overview of methods that have been developed to rectify imbalance so as to make the domains more conducive for multi-class classifiers. Broadly speaking, these methods can be divided into three categories: sampling methods, kernel-based methods, and cost-sensitive methods. In the subsequent sections we will go over each of these categories.

[1]If the classes are highly linearly separable, then clearly this will not be the case. This issue is further elaborated upon in later sections.

2.2.1 Sampling Methods

Sampling methods are concerned with modifying the distribution of instances amongst classes so as to reduce the severity of imbalance. These are classified into two types, namely *undersampling* methods, which reduces the size of the majority class, and *oversampling* methods, which increases the size of the minority class.

Undersampling: Decreasing the majority class

The simplest form of undersampling is random undersampling, where instances from the majority class are randomly removed. However, this can prevent the classifier from learning important concepts. Informed undersampling alleviates the problems of random undersampling by removing only those instances from the majority class that may either be redundant, noisy or borderline. The method involves selecting a random majority class instance and adding it to the minority class. Then, the 1-NN rule is applied to the majority class, using the minority class for selection. Only misclassified instances are added to the minority class to construct a new training set. The aim of this procedure is to remove redundant instances from the majority class. The next step involves removing *Tomek Links*, which are described as follows: Let x be an instance from the minority class, and y be an instance from the majority class. If no z exists such that $\delta(x, y) > \delta(x, z)$ or $\delta(x, y) > \delta(y, z)$, then $x - y$ is a Tomek Link, where δ is a distance function. This step removes instances that might either be noisy or borderline. The final majority class is then added to the minority class and the resultant dataset is used for learning a classifier.

Oversampling: Increasing the minority class

As with undersampling, the simplest oversampling method is random oversampling, where minority class instances are randomly replicated. However, replicating instances from the minority class can lead to overfitting, and thus it is prudent to employ informed oversampling. The most popular informed oversamping method is SMOTE [14] (Synthetic Minority Oversampling Technique). Instead of performing

random replication, SMOTE generates synthetic minority class instances using interpolation between existing instances. The idea is as follows: first, k nearest neighbours (generally set to 5) of a minority class instance x are selected. Let us denote this set by K. Then, a subset $S \subset K$ is selected. Synthetic examples are then generated between each each $s \in S$ and x, and are calculated by taking the difference $\vec{x} - \vec{s}$ and multiplying this difference by a random real between 0 and 1. This approach prevents overfitting as it causes the decision boundaries induced for the minority class to expand into the majority class instance space, thus making the decision boundary more general.

Subsequent to the introduction of SMOTE, researchers have proposed several variants of the original algorithm. Chawla *et. al.* in [15] apply SMOTE within a boosting framework. Specifically, in SMOTEBoost, during each round of boosting, SMOTE is applied to the minority class. This increases the diveristy of the ensemble, as each iteration produces a different set of synthetic samples. Two other variants, SMOTE-Tomek and SMOTE-ENN, are introduced in [4]. In the former, after the application of SMOTE, Tomek links are removed. The goal is to remove noisy or borderline synthetic samples. SMOTE-ENN uses the Edited Nearest Neighbour as a post-processing step. In ENN, if a majority class instance is classified incorrectly by its three nearest neighbours, it is removed. On the other hand, if a minority class instance is misclassified, then the majority class instances from its three nearest neighbours are removed. Both variants were demonstrated to perform very well in situations with very few minority class instances. [30] introduce two variants, Borderline-SMOTE1 and Borderline-SMOTE2, that generate instances only in the border regions. The first variant only generates between borderline minority neighbours, whereas the second variant generates instances between majority neighbours too, but shifts the synthetic samples to be closer to the minority instance.

Wallance *et. al.* conduct a study in [82] to analyze the performances of undersampling and oversampling. They highlight that undersampling is a high variance strategy, as classifiers produced over different bootstrap samples will have different performances. To rectify this, they propose to use bagging with random undersam-

pling to reduce the variance, thus building an ensemble of classifiers, each of which have been trained on a unique undersampled set. It is argued that as SMOTE only generates samples within the convex hull of the minority class space, it fails to reduce the bias of the learner over the imbalanced dataset; this is particularly the case when the training set is very small and the imbalance is severe. In such situations, they show that bagged random undersampling outperforms SMOTE.

2.2.2 Kernel Based Methods

Kernel-based techniques, developed from statistical learning theory, have also been developed to handle imbalanced datasets. In regular SVMs, the optimal separating hyperplane is biased towards the majority class in order to minimize error rates. In the work of Akbani *et. al* in [1], different errors costs are used for different classes so as to shift the decision boundary away from majority class instances. Kang *et. al.* in [43] do not modify the actual SVM classifier; instead, they modify the data distributions and use ensembles of SVMs.

Other methods apply boundary alignment techniques to improve the classification in imbalanced domains. Wu *et. al.* in [86] present three different algorithms for adjusting boundary skews. In their later work [87] they propose a kernel-boundary alignment algorithms, based on the idea of modifying the kernel matrix based on the imbalance in the data distribution. A total-margin based adaptive fuzzy SVM kernel method is proposed by Liu *et. al* in [56], which has different cost algorithms that allow it to self-adapt to different cost distributions. Furthermore, training examples are treated differently according to their relative importance.

2.2.3 Cost-Sensitive Methods

Cost-sensitive methods tackle the imbalance problem by taking into account the costs associated with misclassifications. At the core of this framework is the *cost matrix*, which describes the various misclassification costs associated with each class. Typically, and as one would expect, the costs for misclassifying a minority class instance

are much higher than the costs for misclassifying a majority class instance. Using these cost matrices, cost-sensitive learning aims to minimise the overall misclassification cost.

One of the more popular approaches in the cost-sensitive framework involves modifying AdaBoost to handle costs in the weight update strategy during the training iterations [23, 24]. In essence, these methods increase the probability of sampling a costly instance (*i.e.* a minority class instance) during each training iteration, allowing classifiers to target the minority class space more often. A major drawback with these methods, however, is that they all assume the availability of a cost matrix, an assumption that, more often than not, is impractical in practice.

Alternatively, neural networks have also been adapted within the cost-sensitive framework [52]. One approach involves altering the outputs during the training phase to bias the network to focus on the minority, or expensive, class. Furthermore, the learning rate can also be modified during the weight updates, such that costly examples have a greater impact on the learning rate change as compared to other examples. The purpose of this is to focus more attention on costly examples during learning by decreasing the rate for each corresponding costly example.

2.3 Imbalanced Learning with One-Class Classification

The previous section provided an overview of various methodologies that are used to counter imbalance inherent in many domains, the goal being to make the domain suitable for learning with multi-class classifiers. However, there are cases when learning is only possible with a single class (this is explored in greater detail in Chapter 3). This section will introduce the paradigm of one-class (OC) classification. We will begin with providing an overview of one-class classification, followed by a description of the popular one-class classifiers that are used in practice.

2.3.1 An Introduction to One-Class Classification

Many real-world situations are such that it is only possible to have data from one class, the *target class*; data from other classes, the *outlier classes*, are either very difficult or impossible to obtain. Another case of application arises when there is the issue of novel classes; a multi-class classifier can be constructed using the data from available classes, however, it might fail when presented with data from novel classes that were not previously considered. Examples of such domains include those in which there are almost an infinite number of instances from the outlier set and novel classes can arise, such as in typist recognition, or those in which obtaining instances from the outlier classes is dependent upon the occurrence of a *rare event*[2], such as the detection of oil spills [51] and the inclusion of journal articles for systematic reviews [60].

A traditional approach to one-class classification is to use density estimation. This is performed by attempting to fit a statistical distribution to the data from a single class (the target data), and using the learnt density function to classify instances as belonging either to the target class (high density values), or to the outlier class (low density values). Parametric approaches rely on reliably estimating the distribution of the data beforehand, a challenging and impractical task given that most real-world data takes a complex distribution. An alternative approach to parametric techniques would be to use non-parametric techniques, such as Parzen Windows [20]. But, as the dimensionality of the data increases, these methods suffer from the well known curse-of-dimensionality problem, whereby the computational complexity for density estimation increases drastically.

There are certain algorithms that have been designed specifically for, or can be employed for, OC classification. A variant of a neural network that has been utilized for one-class classification is the autoassociator (AA), which can be thought of as a compression neural network, where the aim is to try to recreate the input at the

[2]It is likely that the outlier class for classification is the target class in reality. However, we use the term *target class* to denote the *majority class*, and it may or may not be the intuitive target class.

output, with the compression taking place at the hidden layers. Hempstalk *et al.*, in [33], describe a method for estimating the probability density function of a single class by first obtaining a rough estimate of the density of target class, generating an artificial class based on it and then performing binary learning. Yet another example of a OC classifier is the OC Support Vector Machine (OCSVM) [58]. OCSVMs assume the origin in the kernel space to be the second class, and, subsequently, learn a boundary that separates the target class from the origin. In subsequent sections, we will go over these algorithms in greater detail.

Autoassociator

Autoassociators [37, 40] are similar in architecture to a multi-layer perceptron. A feedforward neural network is applied to supervised binary learning, in which an input vector x is fed to the input layer of the network, and the output at the output layer is a binary value y, for instance $y \in 0, 1$, that signifies the classification of x. The learning is supervised since for each training instance supplied to the network during the training phase, its classification is known beforehand. Thus, after training, it is expected that the network will be able to correctly (more realistically, correctly within an acceptable error range), classify unknown instances by assigning them the appropriate classification labels. An autoassociator, on the other hand, deals with unsupervised learning. Figure 2.1 illustrates the general architecture of an autoassociator. For autoassociators, the network is trained to reproduce the input at the output layer.

Binary classification takes place as follows: instances from only a single class are presented to the network for training purposes. Thus, the network learns to reproduce only instances from the supplied class at the output layer. Consequently, the reconstruction error between the input and the corresponding output is expected to be very small. If instances from an outlier class (i.e., any class other than the one used to train the network) are presented to the network, the reconstruction error is expected to be relatively large. If a threshold can be set on the reconstruction error such that instances with errors below it are classified for the target class, and those

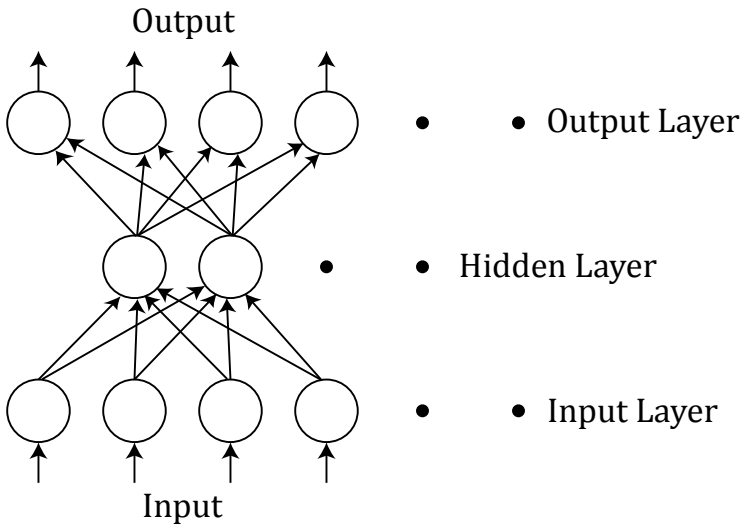


Figure 2.1: Architecture of an Autoassociator

with errors above it are classified as belonging to the outlier class, binary classification can be done. The reconstruction error is calculated as shown in Eq. 1,

$$Error = \sum_{i=1}^{d} [x_i - y_i]^2 \tag{2.1}$$

where d denotes the dimensions of the instance space, x is the input vector and y is the output at the output layer, i.e. the reconstruction of x by the autoencoder. Training is done using the backpropagation method commonly used for feedforward neural networks.

The authors in [40] investigate the employment of autoassociators for one-class classification. Tests are done on three different domains; one involving the recognition of faulty helicopter gearboxes based on the whining noise emitted from them during operation, the second involving the recognition of promoter genes, and the third involving discriminating between rocks and mines based on sonar signals. It was found that the autoassociator outperformed the neural network on the second and third domains, while being comparable to the neural network on the first domain.

Density and Class Probability Estimation

Yet another method developed to deal with one-class learning is one which combines estimating the probability density function and class probability functions (PDEN). This method was originally motivated to deal with the problem of typist recognition, where the task is to identify a user based on typing patterns. In this scenario, the number of classes is extremely large (the largest possible number would naturally be the number of humans with two hands). Thus, this illustrates another use of one-class learning, and that is in the case when obtaining data from classes other than the target class is either impossible or extremely difficult.

This method works by first estimating, roughly, the probability density function of the target class using a density estimator. Once the density is estimated, it is used to generate an artificial class, typically with the same number of instances as in the target class. The result is that there are now two distinct classes. The problem is now transformed into a binary classification problem. A standard binary classifier is then applied to the new dataset to obtain the class probability function (i.e., what is the probability, given an instance, that it belongs to a particular class). These approximations are then used to estimate the probability density function (pdf) of the target function. It is important to note that the approximation of the pdf in the first step is very rough; the approximation obtained in the final step is expected to be much closer to the actual pdf.

The pdf is derived using the information described in the paragraph above using Eq. 2,

$$P(X|T) = \frac{(1 - P(T))P(T|X)}{P(T)(1 - P(T|X))}(P(X|A) \tag{2.2}$$

where X is an instance, T is the target class and A is the artificial class. Clearly, $P(X|T)$ is the desired pdf of the target class, $P(T)$ is the probability of the target class (in case the number of instances of the artificial class is the same as that of the target class, this would be 0.5), $P(T|X)$ is the class probability function of the target class learnt by the applying a binary classifier to the target and artificial classes and $P(X|A)$ is the pdf of the artificial class (as mentioned above, this is a

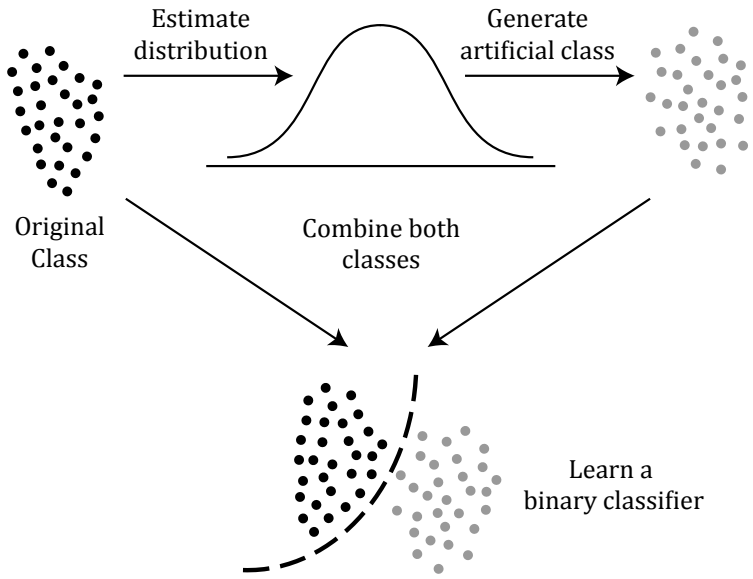


Figure 2.2: Density and Class Probability Estimation

crude approximation of $P(X|T)$). The various steps used for deriving $P(X|T)$ are illustrated in Figure 2.2. When used for the typist recognition problem and other UCI datasets, it applied bagged pruned decision trees as the class probability estimator method, and showed improvement over both a density estimator to estimate the density function, and the class probability estimator (bagged pruned decision trees) trained on the semi-artificial two class problem.

One-Class Support Vector Machines

One-Class Support Vector Machines (OCSVMs) [68] are a modification of the standard binary SVMs in that they treat the origin of the transformed feature space as the outlier class. Essentially, the problem is framed as follows: Assume that a dataset comes from a probability distribution P. Find a subset S of the dataset such that the probability that an instance x from P lies outside of S, *i.e.* $p(x \notin S)$, is bounded by some *a priori* value $v \in (0, 1)$.

The One-Class SVM returns a function f that is positive on instances belonging

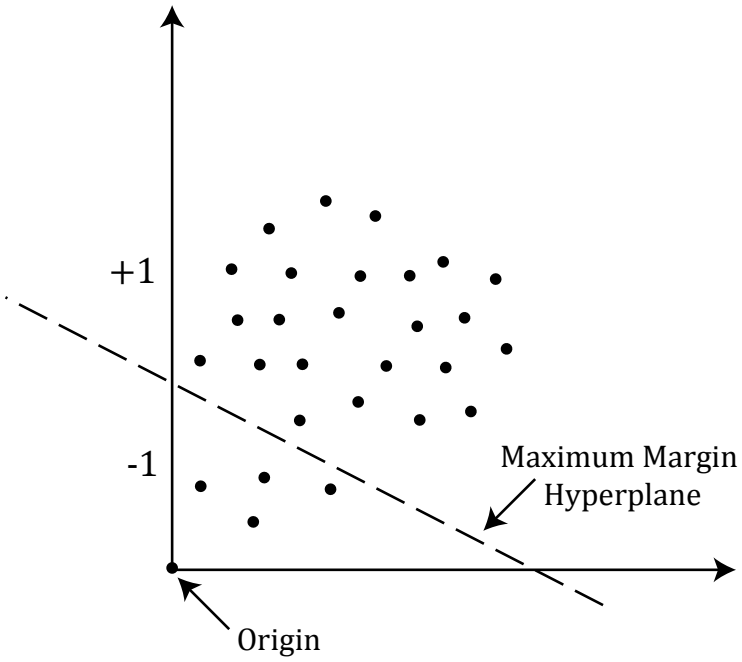


Figure 2.3: The One-Class Support Vector Machine

to S, and negative on those belonging to the complement of S:

$$f(x) = \begin{cases} +1 & \text{if } x \in S \\ -1 & \text{if } x \in \overline{S} \end{cases} \tag{2.3}$$

In other words, the algorithm for One-Class SVM generates a function f that returns $+1$ in a region S capturing the vast majority of instances (the target class), and returns a -1 in the rest of the region. This approach is illustrated in Figure 2.3.

Tax *et. al.* in [79] propose a one-class classifier in a similar vein, called the Support Vector Data Description (SVDD). The classifier aims to learn a hypersphere around the target data; the volume of the hypersphere is minimized in order to minimize the likelihood of accepting outliers.

2.3.2 Other Methods for Outlier Detection

Apart from the algorithms discussed in the aforementioned sections, other methods exist that can be utilised for outlier detection. In this section, we will briefly review

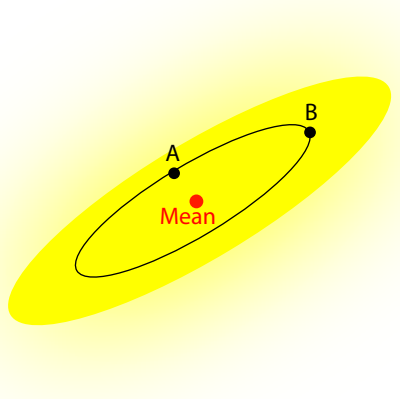


Figure 2.4: Illustration of the Mahalanobis Distance between two points **A** and **B** from the mean. Both points have the same Mahalanobis distance, but different Euclidean distances from the mean.

some of these methods.

Statistical Methods employed for detecting outliers in a dataset start with the assumption that the probability distribution function (pdf) of the dataset is known, and can be estimated. Outliers are then discovered as having a very low likelihood of being generated by the pdf. The *Mahalanobis Distance* (MD), for instance, can be utilized for detecting outliers. The Euclidean distance offers no information with regards to where points lie within the distribution of interest. The MD is a parametric distance-measure used for Gaussian distributions. Intuitively, if one visualizes a Gaussian distribution as being composed of hyperelliptical contours, the MD gives the distance with respect to the position of instances within these contours; specifically, it discriminates between instances lying in different density regions of the distribution. Two points that might be within the same Euclidean distance from the mean might have a very different MD, as one may lie in a low density region, and another may lie in a high density region, and vice versa. Figure 2.4 illustrates the idea of two points having the same MD from the mean (*i.e.* they lie on the same hyperelliptical contour representing a region of equivalent density) but different Euclidean distance.

The calculation of the MD involves knowing the mean μ and the covariance matrix Σ of the distribution. Once these are known, the MD of an instance x from the mean μ is calculated as:

$$MD(x, \mu) = (x - \mu)^T \Sigma^{-1} (x - \mu) \tag{2.4}$$

Clearly, the larger the MD between an instance and the mean, the lower the likelihood of that instance being generated by the underlying pdf. In this manner, a threshold set on the MD can be used for one-class classification.

Depth-Based approaches to outlier detection [49] are applied on the assumption that the outliers are present at the border regions of the data space. In order to determine outliers, the data is organised into layers of convex hulls; outliers are those instances that lie on the outermost layers. However, these approaches are only practical in at most 3-dimensional spaces.

Distance-Based approaches discover outliers based on their distance to their neighbours; they assume that outliers will be much further apart from their neighbours, and will consequently have a sparse neighbourhood, as opposed to a dense neighbourhood for normal instances. A standard approach given by Knorr *et. al.* in [44] is as follows: Given a radius r and a percentage p, an instance x is an outlier if not more than $p\%$ of instances from the dataset D have a distance less than r. The set of outliers X, from D, is thus calculated as:

$$X = x|\frac{|d \in D|\delta(d,x) < r|}{|D|} \leq p \tag{2.5}$$

As the dimensionality of the data increases, methods that rely on distances and neighbourhoods become impractical. The *Angle Based Outlier Degree* by Kriegel *et. al* [48] is a method that can be applied in high-dimensional space, as angles are more stable than distance measures in high dimensions. The Variance in Angle Spectrum (VAS) [48] uses the variance in the cosine of the angles between an instance p and all other instances $x \in D$, where D is the set of all instances. An outlier is identified as having a low variance amongst the angles. This is based on the rationale that an outlier will most likely lie in a border region of the distribution, and therefore most points will lie in the same direction. A non-outlier point, on the other hand, will be in the middle of the distribution, and thus most points will surround it with equal density, thereby giving a high variance in the angle. The variance for every point

$p \in D$ is calculated as:

$$VAS(p) = var_{x,y \in D} \left(\frac{\langle \vec{xp} \cdot \vec{xp} \rangle}{\|\vec{xp}\| \cdot \|\vec{yp}\|} \right)$$

A low value of $VAS(p)$ will imply that p is most likely an outlier, and vice versa. This approach, however, is computationally expensive, with an $O(n^3)$ complexity.

2.4 Challenges in Imbalanced Domains

Previous sections outlined the two different paradigms for dealing with imbalance. Sampling methods aim to directly modify the distribution of the dataset, aiming to either increase the size of the minority class or reduce the size of the majority class. Kernel Based methods either modify the kernel directly, or are integrated with sampling techniques. Cost-sensitive methods take into account the costs for misclassification to handle imbalance. One-class classifiers, on the other hand, learn over a single class only.

Real-world domains are seldom "simple" enough for all the various methods to be applied directly; there are various theoretical considerations that need to be understood. Apart from just imbalance, there are other factors that can impact the performance of classifiers. Stefanowski [74] highlight that the presence of small disjuncts in the minority class, overlap between the classes, borderline and noisy examples as well the presence of examples from the minority class in the majority class, all have a greater influence on the induction process than just the level of imbalance between the classes. Thus, it is prudent to understand these factors in conjunction with class imbalance, and in this section, we review research done within the field of imbalance to understand and manage these complications.

We begin by reviewing work done in understanding the relationship between level of imbalance and the level of overlap between various classes in domains. Apart from overlap, complexity in domains can also arise from multi-modality; this is reviewed in the subsequent subsection.

2.4.1 Imbalance and Overlap

While much of the focus in literature has been on rectifying the imbalance explicitly, a few researchers have quested the extent to which imbalance really affects classifier performance. Are there properties of the domain that might make the impact of imbalance less severe? Japkowicz *et. al* in [38] discovered that if the classes are linearly separable, a classifier such as a support vector machine (SVM) would not be impacted by imbalance. Specifically, they note that the problems due to class imbalances are relative to various factors, such as the complexity of the concept (*i.e.*, how complex the probability distribution function that generates the data is) and the size of the training set; the classifier performance will be immune to imbalance if the concept is simple and the training set is large.

Prati *et. al* conduct a similar study in [64]. Using 10 artificially generated datasets, each with varying levels of overlap, they conduct classification experiments over each with increasing degrees of imbalance. They observe that as the overlap decreases, the effect of imbalance becomes less severe. To deal with overlap, they propose two alternatives to SMOTE, SMOTE-Tomek and SMOTE-ENN (discussed earlier in this chapter) in [4], and show that when dealing with imbalance and overlap, these methods are able to outperform sampling methods that do not handle overlap explicitly.

In order to further understand the nuances of overlap and imbalance, Garcia in [25] examine the performance of classifiers by looking at imbalance within the overlap region. Specifically, they observe the performance of classifiers when the imbalance level in the overlap region is the same in the rest of the domain space, and when the levels are reversed. Thus, they examine three aspects of data complexity: the overall imbalance ration in the data (global imbalance), the imbalance ration in the overlap region (local imbalance), and the amount of data in the overlap region. They observe that classifiers that focus on local learning, such as k-nn and radial basis functions, are more impacted by local imbalance than global learners such as neural networks, decision trees and Bayesian classifiers. In other words, that class that is

better represented in the overlap regions is better classified by global learners, while the class that is not well represented is classified better by local learners.

All these studies conclude that imbalance is not always the issue; the degree of overlap between the classes has a major impact. This dependency between overlap and imbalance is further studied in [18], where they observe that given enough data, imbalance has minimal impact, whereas if the overlap is severe, even optimal classifiers will suffer in performance. Thus, focusing on imbalance alone may not be sufficient, and measures need to considered to handle overlap.

In the following subsection, we consider the other facet of data complexity, that of multi-modality, specifically in the context of one-class classifiers.

2.4.2 Imbalance and Multi-Modality: Divide-And-Conquer

Domains in the real-world are seldom derived from simple distributions; aspects such as dimensionality, availability of data and overlap and multi-modality all impact the ability of classification algorithms to capture the intricacies of the domain. Divide-and-conquer and ensemble learning are typical approaches employed to make learning conducive over complex domains. In this sub-section, we review research conducted in this field. We begin by considering ensemble-based approaches that re-sample the domain and utilize existing ensemble learning algorithms, followed by work done under a clustering-based framework that divides the domain space by some form of clustering. We end this sub-section by considering research done that exploits known domain knowledge in order to divide the domain for improving classification.

Ensemble learning based approaches

Ensemble methods such as bagging and boosting have had tremendous success in multi-class classification, and thus, it should come as no surprise that methods inspired from these algorithms have found their way within the realm of learning with imbalance. Shieh *et. al.* in [72] utilize bagging to create an ensemble of one-class support vector machines (OCSVM). They note that in the presence of noisy and bor-

derline samples, OCSVM learns an enlarged boundary which leads to a large number of false positives. They attribute this to the inherent instability of OCSVM, and thus bagging, a method that improves classification by combining multiple unstable classifiers, is employed to rectify this. They evaluate both a regular OCSVM and the bagged version over artificial and three UCI datastes, and show that they bagged version outperform the regular OCSVM in all cases, especially when they introduce noise into the datasets.

Desir *et. al* employ a different approach for building an ensemble in [19]. Specifically, they create boostrap replicas of the training data, and generate outliers for each set. Thus, they convert each subset of the data into a binary classification problem. Over these subsets, they train random forest classifiers, and aggregate their decision. Through experiments conducted over datasets from the UCI repository, they compare the performance of their method against OCSVM, Gaussian estimator, Parzen windowing and Mixture of Gaussians, and demonstrate that on most datasets, their approach performs equally well or better than these algorithms.

Clustering-based approaches

The seminal work of Jacobs [35] introduces the idea of using an ensemble of neural networks in a divided data space. Specifically, they divide the training data into a number of subsets, and train neural networks on each of these subsets, in essence building a mixture of expert neural networks, which in turn are supervised using a gating network which decides which expert should be used for which training case. Masoudina *et. al.* [59] divide the methods in this framework into two types: The *Mixture of Implicitly Localized Experts* methods stochastically partition the data space into a number of smaller sub-spaces by utilising a special error function. Experts are then specialised in each of these subspaces. Alternatively, the *Mixture of Explicitly Localised Experts* partition the data space prior to training, and then assign networks to each sub-space.

Within an unsupervised learning paradigm, the later partitioning framework can be considered as an approach to improve classification. One of the earliest works on

employing a divide-and-conquer paradigm is presented in [78], where they study the impact of combining one-class classifiers to model complex domains. In particular, they employ four different one-class classifiers, the support vector data description (SVDD), the autoassociator (AA), k-means and k-centre methods, and create ensembles by combining different classifiers over the same dataset, and by training the same classifier over different feature subspaces. The domain they consider is the handwritten digits dataset, and it has six sets of features: profile features, Fourier features, Karhunen-Loeve features, morphological features, pixel features and Zernike features. The results from the ensemble are combined via six different methods: mean, weighted mean, product of weighted votes, mean and product of estimated probabilities. They observe that while the performance of combining different classifiers is below the performance of individual classifiers, combining the same classifier trained over the different feature subspaces is far more effective.

The work of Tax described above focused on ensembles created over different feature subspaces. At the other end of the spectrum, research has been conducted in employing clustering for inducing more accurate classifiers, specifically as a means for resolving the issues arising from *small disjuncts*. As stated in previous sections, small disjuncts arise as a result of *rare cases* in the data space, a form of within-class imbalance. The issues related with small disjuncts are exacerbated in domains that exhibit high complexity [38], *i.e.*, the underlying probability distribution is complex. A method proposed in [42] employs clustering alongside random oversampling to rectify both within-class and between-class imbalances in an imbalanced binary problem; clustering is used as an aid to identify sparse regions in the data space that have a high likelihood of leading to the problem of small disjuncts. The method works as follows: a clustering algorithm, in this case k-means, is used to cluster both the majority and minority classes separately. Once the clusters are created, the majority class clusters are randomly oversampled to have the same size as the largest cluster in the majority class. Then, the minority class clusters are randomly oversampled until there are ${}^{maxclasssize}/_{Nsmallclass}$ elements in each cluster, where $maxclasssize$ is the size of the largest cluster, and $Nsmallclass$ are the number of clusters created in

the minority class. They report, based on tests on artificial and UCI domains, that cluster-based oversampling outperforms random oversampling and other methods for dealing with imbalance and the problem of small disjuncts, particularly when the domain has a small size and high complexity, as using clustering identifies rare cases and resamples them individually, thus avoiding the creation of small disjuncts in the generated hypothesis.

Another approach of using clustering is presented in [41], in which a multi-class problem is clustered, and each cluster is treated as a separate class. Specifically, each cluster is labelled as its own class, and a supervised learning method is then applied to the multi-class problem. The approach is repeated with a different number of clusters, and the results are combined in a decisive vote. They report results on five domains from the UCI repository, and observe that in three of the domains, there is a marked improvement in performance, whereas in the other two, there is no change. In [83], Wang *et. al.* exploit the inherent target data structures obtained via hierarchical clustering to create an ensemble of spherical one-class classifiers. Further work in employing clustering for creating one-class classifier ensembles is conducted in [55], where the authors employ the k-means algorithm to create an ensemble of OCSVM classifiers. A general purpose framework for employing clustering for improving one-class classification is proposed in [47]. Their framework consists of three parts: the choice of clustering algorithm, the choice of one-class classifier and the choice one how to combine decisions. In all studies, the ensemble appears to outperform the single one-class classifiers. Within the area of network intrusion detection, Leung *et. al* in [53] develop a clustering algorithm based on pMafia to perform anomaly detection on network data, with the aim to cover 95% of the training data (using the KDD Cup dataset), thus making the 5% of data points not covered as outliers.

Learning with domain knowledge

While the previous sections employed generic methods for partitioning the data space, some research, especially in security, has looked into utilizing the nuances of the domain itself for partitioning the space. In other words, rather than simply cluster

or sample based on a general heuristic, they look at if the groups can be formed such that all the data in a particular group conforms to a pre-determined heuristic as determined by a domain expert.

In the domain of handwritten character recognition, a non-clustering based approach to autoassociation is discussed in [69]. The authors describe *Diabolo Networks*, in which an autoassociator is trained on a particular class, inherently reducing a multi-class problem into multiple one-class problems. Each character is treated as a separate class, and a network is trained on it. The class whose network returns the smallest reconstruction error is assigned to the corresponding test instance. Note that the core idea in this approach may not have to do with alleviating the problem of small disjuncts, but of reducing a complex domain, *i.e.* the set of all alphabets in the English language as applicable to handwritten character recognition, into simpler domains, *i.e.* the individual characters.

In the domain of network security, Giacinto *et. al.* propose a modular system in [27]. Specifically, they observe that traffic over a TCP/IP network is made up of packets pertaining to different services, each characterized by its own unique pattern. Thus, it follows that a unique classifier must be induced for each service, rather than the traffic as a whole. They identify six services: web, mail, ICMP, FTP, ICMP and miscellaneous. Furthermore, for each service, they employ three sets of features, one for content specific information, one for intrinsic information and one related to traffic. Thus, each service has three classifiers trained for it. Using a variety of multi-class classifiers over the DARPA dataset, they authors report an improvement in classification performance and a reduction in false alarms. The employment of multi-class classifiers is not very practical (as attack data is typically unavailable for learning), and thus in a follow up work in [26], the authors extend this framework to train one-class classifiers, specifically, the one-class SVM, parzen density estimator and k-means. As with multi-class classifiers, they observe that the ensemble provides superior performance as opposed to just single classifiers.

Neither of the studies discussed in this sub-section employ a general heuristic. Instead, they divide and conquer based on explicit domain knowledge. The principles,

however, are the same: divide the domain space, and learn over each subset.

2.5 Summary

This chapter provided a brief overview of the imbalance problem and the various techniques described in literature to rectify the problems resulting from imbalance. Sampling methods aim to directly modify the distribution of the dataset, aiming to either increase the size of the minority class or reduce the size of the majority class. Kernel Based methods either modify the kernel directly or are integrated with sampling techniques. Cost-sensitive methods take into account the costs for misclassification to handle imbalance. On the other hand, one-class classifiers learn over a single class only. We further reviewed research undertaken in the field of imbalance to understand the relationship between imbalance and class overlap, as well as studies that look into the construction classifier ensembles, either via clustering, sampling or by using domain knowledge, to improve classification and learning in complex domains.

In Figure 2.5, we provide a graphical overview regarding the various aspects of research in learning over imbalanced domains. We observe that, irrespective of the choice of learning paradigm (multi-class or one-class), the challenges in building efficient classifiers are similar: one has to deal with either overlap, domain complexity (for example, due to multi-modality), or a combination of both.

As the figure demonstrates, and indeed, from the review presented in this chapter, if we are to employ multi-class classifiers, then the task of rectifying the challenges posed by various aspects of imbalance is relatively easy. It is when the choice is to employ a one-class classifier that we observe a dearth in research and solutions for handling all challenges presented by imbalance. The notion of overlap only really makes sense when we have knowledge of both classes; given only data from a single class, it is difficult to ascertain whether other classes will overlap with the known class. This issue will be further exacerbated if the domain exhibits multi-modality. It is this void in research that this thesis aims to fill.

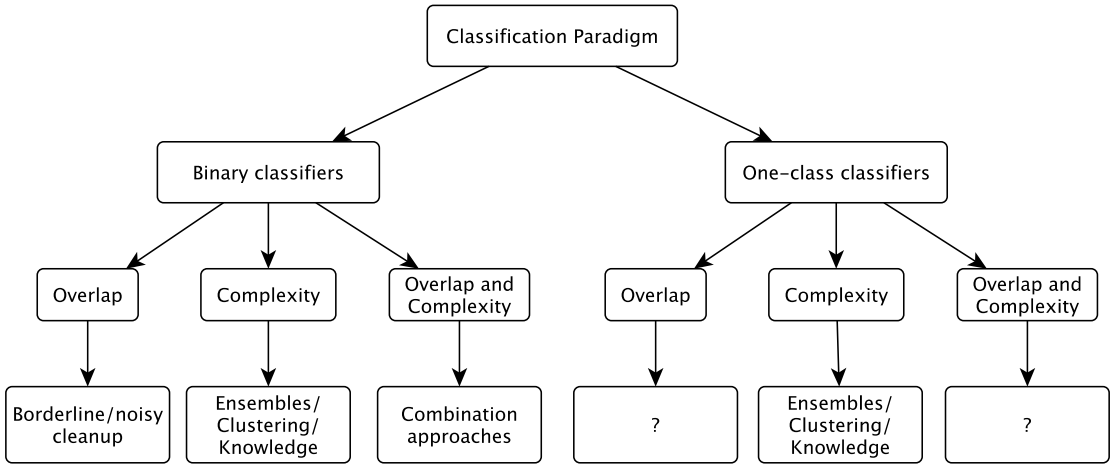


Figure 2.5: An overview of research coverage in the field of learning over imbalanced domains.

Naturally, one can simply counter and say: *Why not just employ multi-class classifiers?* This would appear to make sense; there is substantial work done to handle the various issues of imbalance within multi-class learning. However, we must pause and ask: *Are there situations when we simply cannot employ multi-class classifiers?* If so, then *how do we handle imbalance when presented with complex domains that exhibit an inherent overlap between various classes?* In the following chapters, we provide answers to these solutions. Specifically, in the next chapter, we critically examine the case when employing multi-class classifiers, even when complemented with techniques to manage the imbalance, becomes futile, and the only choice is to utilize one-class classifiers. Subsequently, in Chapter 4, we introduce a framework for learning with one-class classifiers in domains that exhibit overlap and multi-modality.

Chapter 3

One-Class versus Binary Classification

In the previous chapter, we reviewed existing work in learning over imbalanced domains. We introduced the paradigm of one-class classification, along with some of the major classifiers and methods employed within it. However, in most domains, binary classifiers are still the norm, with or without the employment of sampling and other imbalance rectifying methods. Since the use of either of the paradigms is dependent on the level of imbalance inherent in the dataset, a natural question to ask is: at what levels of imbalance does the use of binary classifiers, which rely on discrimination, becomes futile, and using one-class learning, which is based on recognition, becomes the more suitable option? In this chapter, we thoroughly investigate this issue, and establish a case of scenarios where one-class classification is the only suitable paradigm; we first explored this issue in our work in Bellinger, Sharma and Japkowicz [8].

We begin by providing an analysis of the two paradigms within a Bayesian context in Section 3.1. In order to analyze the effects of imbalance, we perform a series of experiments over a multitude of datasets; these are described in detail in Section 3.2. These datasets are inherently suitable for binary classification, and thus, in order to simulate the effects of imbalance, we exponentially decrease the size of the *minority* class, and analyze the effect of this reduction in the performance of various

binary classifiers, with and without the use of sampling methods. We outline the experimental framework in Section 3.3 and present the results in Section 3.4, along with an in-depth discussion in Section 3.5.

3.1 One-Class versus Binary Classification: A Bayesian Perspective

In this section, we discuss in detail the performance of binary classifiers in the context of levels of imbalance in the dataset from a Bayesian perspective.

The Bayes Rule for classification, assuming a zero/one loss function, given two classes ω_1 and ω_2, is: Classify as ω_1 if $p(x|\omega_1)P(\omega_1) > p(x|\omega_2)P(\omega_2)$, else classify as ω_2.

If instances from class ω_1 are much more abundant than those from class ω_2, we will get, for the prior probabilities, $P(\omega_1) >> P(\omega_2)$. Also, given the rarity of instances from class ω_2, the probability density functions (pdf) will be related as $p(x|\omega_1) >> p(x|\omega_2)$. Only in extremely rare, exceptional cases will this inequality be reversed. Given these relationships between the priors and the pdf, using the rule mentioned previously, we observe that we will almost always classify an instance as belonging to class ω_1. Clearly the resulting classifier will be extremely biased towards the majority class and therefore will not be suitable for use in an imbalanced domain.

This analysis shows, from a Bayesian perspective, the effects of imbalance on binary classifiers; they almost always become biased towards the majority class, effectively ignoring the minority class. In contrast, one-class classification ignores prior probabilities since, given a single class ω, the notion of using prior probabilities becomes moot. What we are interested in one-class classification is estimating the pdf of the given target class; once we have that, we can perform classification by imposing

a threshold τ on the value given by the pdf for a given test instance:

$$\text{Classification}(x) = \begin{cases} \text{target}, & \text{if } p(x|\omega) \geq \tau \\ \text{outlier}, & \text{otherwise} \end{cases} \tag{3.1}$$

As we only use information from a single class to build a model (in this case, estimating the pdf), there is no preferential bias present in it. Therefore, one-class classification, from the analysis shown here, becomes the better choice for building classification models when extreme levels of imbalance are present in the data.

The discussion here is from a purely theoretical perspective and does not necessarily relate to any particular classifier. What we are interested in is empirically verifying the analysis presented in this section by running different classifiers on various datasets, and seeing at what point the use of binary classifiers, even when utilized in conjunction with sampling methods, becomes detrimental to the problem at hand.

3.2 Description of the Data Sets

This section provides a description the various data sets used in the experiments. We begin by describing the artificial datasets that we create, followed by the UCI datasets. It should be noted that the datasets described in this section are used in the subsequent chapter as well. In all cases, the class termed "target" represents data that is assumed to be available for learning, and the class termed "outlier" pertains to data that is assumed to be either unavailable or extremely difficult to obtain.

3.2.1 Artificial Data

The purpose of using artificial data is to create an idealized data distribution on which we can concretely test the trends of classifier performance as class imbalance increases. In particular, we use three 5-dimensional artificial datasets which are various combinations of multimodal and unimodal target and outlier distributions. We create two multimodal distributions, one in which there is no overlap between the

modes, and the other in which we force overlap. The specifications for these are as follows (in each $N(\mu, \sigma)$, μ represents the mean vector, and σ the standard deviation) :

Data 1 : Unimodal target and multimodal outlier distributions:

Target : $N([15, 15, 15, 15, 15], 2.75)$

Outlier : $N([5, 5, 5, 15, 15], 2) \cup N([25, 25, 25, 15, 15], 2)$
$\cup N([15, 15, 15, 5, 5], 2) \cup N([15, 15, 15, 25, 25], 2)$

Data 2 : Multimodal target and multimodal outlier distributions, no overlap:

Target : $N([5, 5, 5, 5, 5], 3) \cup N([25, 25, 25, 5, 5], 3)$
$\cup N([5, 5, 5, 25, 25], 3) \cup N([25, 25, 25, 25, 25], 3)$

Outlier : $N([15, 15, 15, 2.5, 2.5], 2) \cup N([27.5, 27.5, 27.5, 15, 15], 2)$
$\cup N([2.5, 2.5, 2.5, 15, 15], 2) \cup N([15, 15, 15, 15, 27.5, 27.5], 2)$

Data 2 : Multimodal target and multimodal outlier distributions, overlap:

Target : $N([10, 5, 5, 10, 5], 3) \cup N([20, 25, 25, 10, 5], 3)$
$\cup N([10, 5, 5, 20, 25], 3) \cup N([20, 25, 25, 20, 25], 3)$

Outlier : $N([17.5, 15, 15, 5, 2.5], 2) \cup N([25, 27.5, 27.5, 17.5, 15], 2)$
$\cup N([5, 2.5, 2.5, 12.5, 15], 2) \cup N([12.5, 15, 15, 15, 25, 27.5], 2)$

Figure 3.1 shows the principle component analysis (PCA) plots of the first three components of the unimodal artificial dataset, whereas Figures 3.2 and 3.3 display the multimodal artificial distributions without and with overlap, respectively. In all datasets, there are 4000 target and 2000 outlier instances (equally split between each mode).

3.2.2 UCI Datasets

Apart from the artificial datasets, we also consider datasets from the UCI repository [54], each with its own unique characteristics. Table 3.1 lists the datasets used, along

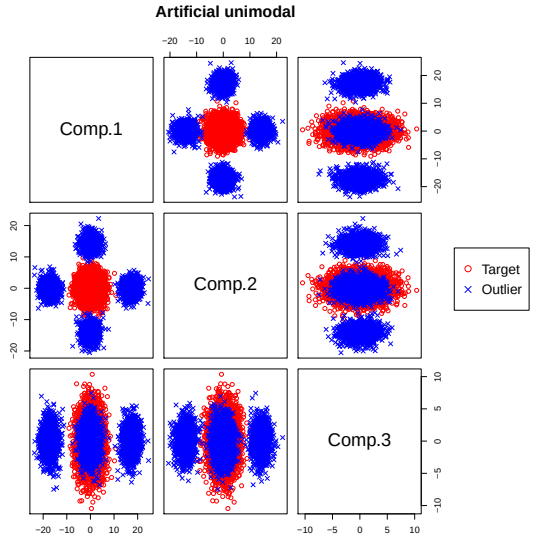


Figure 3.1: The first three principal components of the *unimodal artificial* dataset. The target class exists as a unimodal gaussian, whereas the outlier class has four modes.

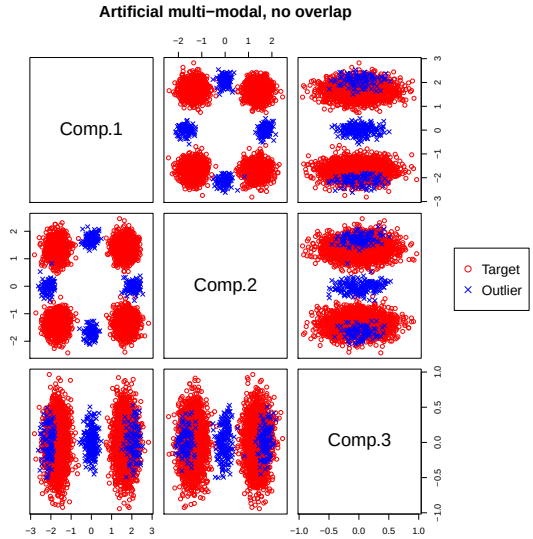


Figure 3.2: The first three principal components of the *multimodal artificial* dataset with no overlap. Both the target and outlier classes exist as distributions with 4 modes with minimal overlap between each class.

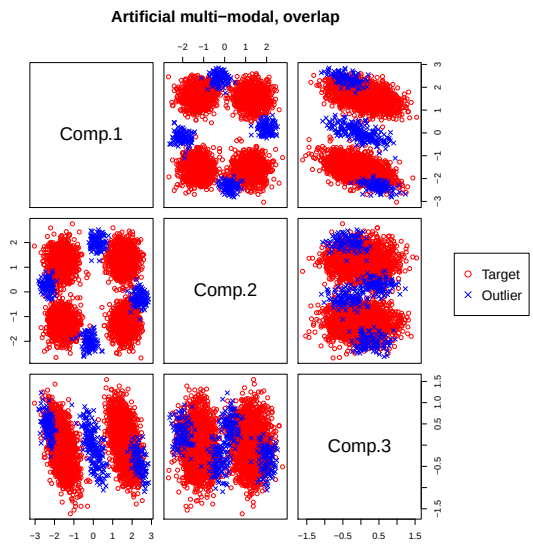


Figure 3.3: The first three principal components of the *multimodal artificial* dataset with overlap. Both the target and outlier classes exist as distributions with 4 modes with significant overlap between each class.

with the initial number of target instances and outlier instances in each dataset (prior to the exponential increase in the level of imbalance). All the datasets have numeric attributes and no missing values. As can be seen, with perhaps the exception of the thyroid disease dataset, the initial ratios indicate that there is not nearly enough balance to warrant using one-class classification.

In the following section, we describe the dataset and look at them in terms of their first three principal components. All the datasets are real-valued and have

Table 3.1: Description of the UCI datasets.

Dataset	Number of Targets	Number of Outliers
Diabetes	500	133
Heart Disease	150	60
Ionosphere	225	63
Thyroid Disease	3541	115
Sonar	111	97
Alphabets	3077	1538
Forest	16500	6499
ForestC1	2500	1249
ForestC2C5	2400	1200

no categorical variables. The purpose of this is to obtain an understanding as to the general distribution properties of the data. Given that the alphabets dataset and forest cover derivatives have a very large number of instances, the outliers in the PCA plots are displayed with very low opacity in order to make the plots more comprehensible. For completeness, the plots with complete opacity are shown in Appendix A

The *Diabetes* dataset

The dataset has 8 dimensions (excluding the class attribute) and contains records for females of Pima Indian heritage that are at least 21 years old. The class corresponds to whether or not the patient has diabetes or not.

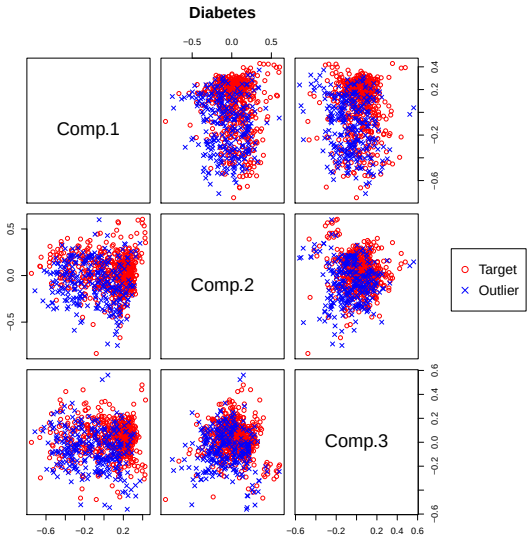


Figure 3.4: The first three principal components of the *Diabetes* dataset.

The scatterplot matrix for the first three principal components are shown in Figure 3.4. The diabetes dataset exists primarily as a unimodal distribution. However, we note that there is a significant overlap between the two classes. This property, as expected, exacerbates the induction of an accurate classifier.

The *Heart Disease* dataset

The dataset has 13 dimensions (excluding the class attribute) and contains patient records indicating the presence or absence of a heart disease.

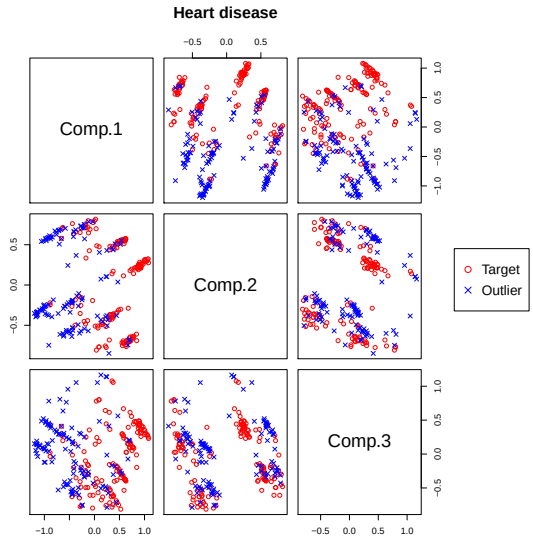


Figure 3.5: The first three principal components of the *Heart Disease* dataset.

The scatterplot matrix for the first three principal components are shown in Figure 3.5. This dataset is highly multimodal, with tight clusters spread throughout the domain space. At the same time, however, we note that the level of overlap is not as severe.

The *Ionosphere* dataset

The dataset has 34 dimensions (excluding the class attribute). The data is radar data that was collected at a station in Goose Bay, Labrador. The classes correspond to "good" radar returns (show evidence of some structure in the ionosphere) or "bad" radar returns (no evidence of any structure).

The scatterplot matrix for the first three principal components are shown in Figure 3.6. The domain appears to be primarily unimodal, with a high level of spread in the first two components. Furthermore, there is significant overlap in the first two components as compared to in the first and third, and the second and third

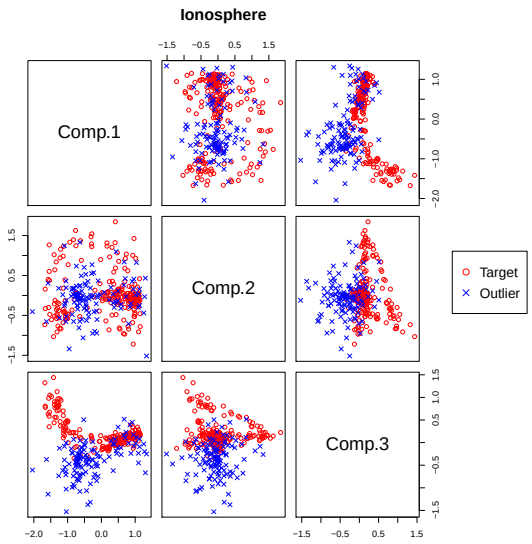


Figure 3.6: The first three principal components of the *Ionosphere* dataset.

components.

The *Thyroid Disease* dataset

The dataset has 33 dimensions (excluding the class attribute) and contains thyroid disease records supplied by the Garavan Institute. The class attribute corresponds to the presence or absence of a Thyroid disease.

The scatterplot matrix for the first three principal components are shown in Figure 3.7. While both the target and outlier classes in this domain have a dense cluster where the majority of the data is present, the taget class is much more spread than the outlier class. Furthermore, the outlier class is almost completely contained within the target class.

The *Sonar* dataset

The dataset has 60 dimensions (excluding the class attribute) and contains sonar signals bounced off a metal cylinder and a roughly cylindrical rock, each of which corresponds to a unique class.

The scatterplot matrix for the first three principal components are shown in Figure 3.8. The data from both classes is fairly uniformly spread across the data space for

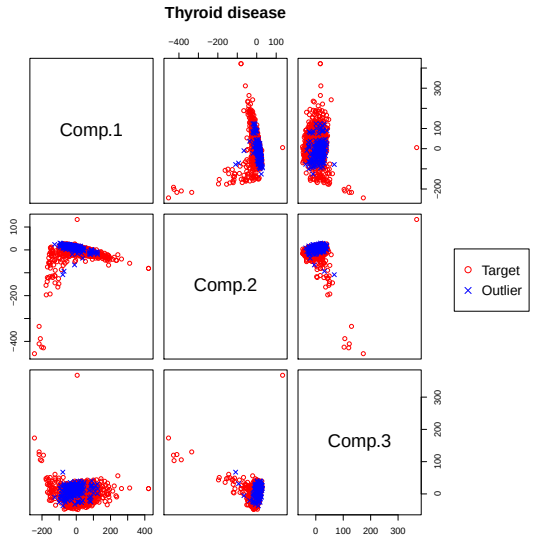


Figure 3.7: The first three principal components of the *Thyroid Disease* dataset.

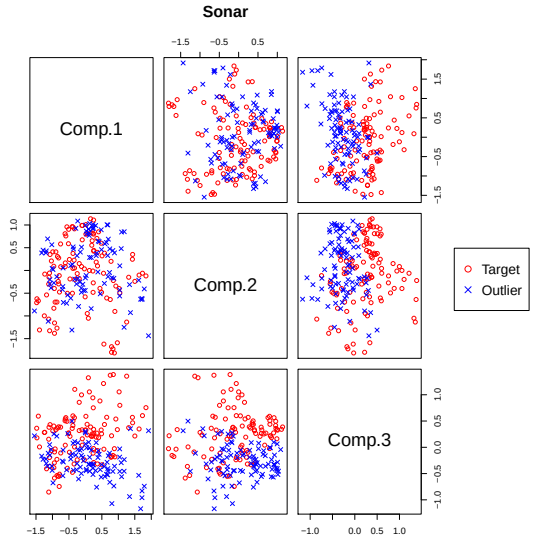


Figure 3.8: The first three principal components of the *Sonar* dataset.

all three components. Both classes also exhibit a significant degree of overlap. There are 60 dimensions in the domain.

The *Alphabets* dataset

The dataset has 16 dimensions (excluding the class attribute) and contains numerical attributes that describe one of the 26 capital letters in the English alphabet. There are thus 26 classes, one for each attribute. In order to convert it into a binary class problem, we represent the target class by all instances corresponding to the letters **I**, **J**, **M** and **N**; all other letters constitute the outlier class. The pairs of **I** and **J**, and **M** and **N** are typically more similar to each other than to other alphabets (a point that is verified by the fact that most misclassifications for each of these letters happen with the other in the same pair for certain classifiers). Furthermore, these pairs turn the resulting distribution into a multi-modal distribution with three modes.

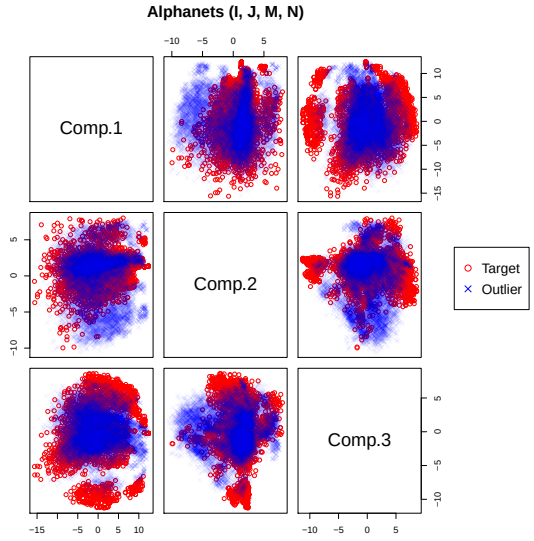


Figure 3.9: The first three principal components of the *Alphabets* dataset.

The scatterplot matrix for the first three principal components are shown in Figure 3.9. The most obvious feature of the target class is the presence of three modes: one large and two smaller ones. We also note that the outliers almost completely cover the larger of the three modes; the two smaller modes lie outside the outlier space. If

we are to examine the first two components, we observe that the outlier space extends outside the larger mode. The number of dimensions is 16.

The *Forest Cover* dataset

The dataset has 54 dimensions (excluding the class attribute) and contains cartographic variables that describe the forest cover in a 30×30 meter cell. There are seven different classes, and we construct three datasets from the original, based on the following information (taken from the website at [66]):

> As for primary major tree species in these areas, Neota would have spruce/fir (type 1), while Rawah and Comanche Peak would probably have lodgepole pine (type 2) as their primary species, followed by spruce/fir and aspen (type 5). Cache la Poudre would tend to have Ponderosa pine (type 3), Douglas-fir (type 6), and cottonwood/willow (type 4).
>
> The Rawah and Comanche Peak areas would tend to be more typical of the overall dataset than either the Neota or Cache la Poudre, due to their assortment of tree species and range of predictive variable values (elevation, etc.) Cache la Poudre would probably be more unique than the others, due to its relatively low elevation range and species composition.

The datasets are described in detail below:

Forest dataset This is composed of types 3, 4, 6 and 7 as the target class. While types 3, 4 and 6 form their own unique niche in Cache la Poudre, the resulting distribution was found to be highly simple to learn. As a result, to add an element of complexity and multi-modality, we combined the three with type 7 (Krummholz).

The scatterplot matrix for the first three principal components are shown in Figure 3.10. The target class exists as a bimodal distribution. Specifically, the larger mode corresponds to cover types 3, 4 and 6, where as the smaller mode is cover type 7. The outlier class almost completely contains the target concept,

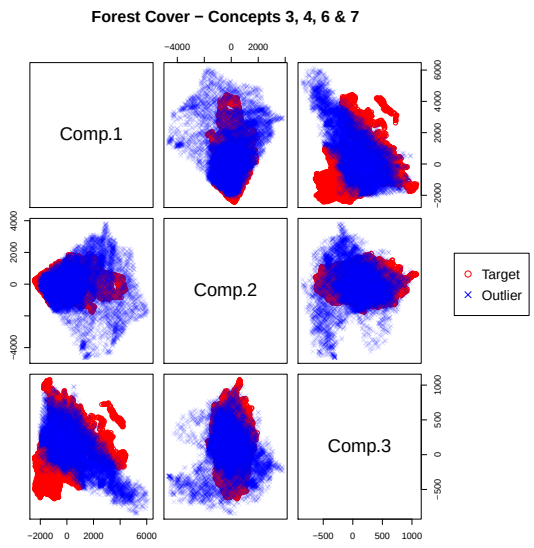


Figure 3.10: The first three principal components of the *Forest* dataset.

except in the first and third components; the target concept in these spreads outside the outlier space.

ForestC1 Type 1 is found in Neota, is the second largest of the classes, and, as described above, while not as distinct as the species in Cache la Poudre, is still atypical of the species found in the region. Thus, the second forest cover dataset had only Type 1 as the target class.

The scatterplot matrix for the first three principal components are shown in Figure 3.11. As expected, the target class exists as a unimodal distribution. Furthermore, the outlier class almost completely overlaps the entire target class.

ForestC2C5 The final dataset represents the types 2 and 5, which are the largest distinct group in the dataset, and the most typical of the region.

The scatterplot matrix for the first three principal components are shown in Figure 3.12. The final forest dataset is comprised of cover types 2 and 5 as the target class. Once again, as expected, given the similarity of the two types, the target distribution appears as a unimodal distribution. Unlike the previous dataset, however, the level of overlap between the target and outlier distributions is must less severe.

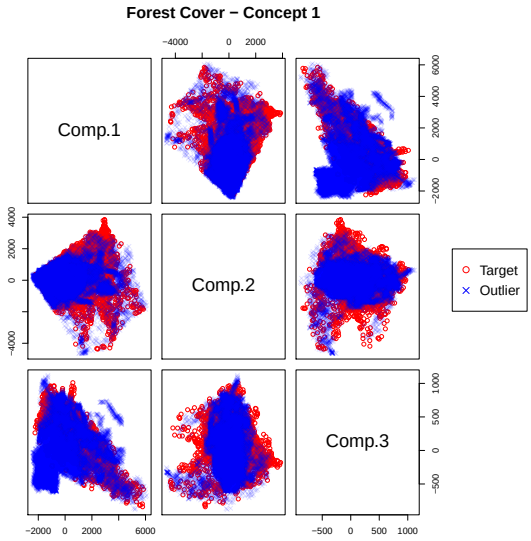


Figure 3.11: The first three principal components of the *ForestC1* dataset.

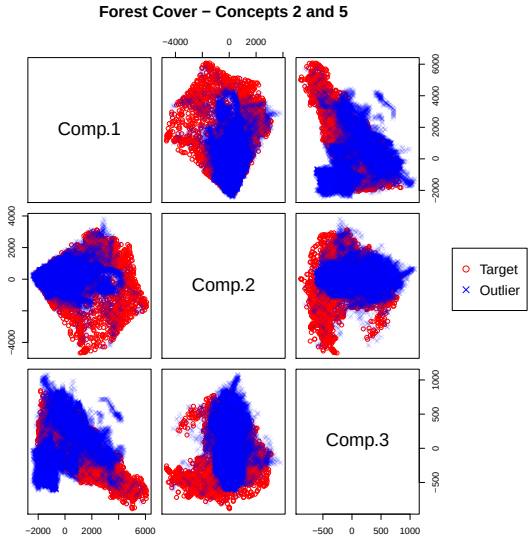


Figure 3.12: The first three principal components of the *ForestC2C5* dataset.

3.3 Experimental Framework

In order to perform a comprehensive analysis into the applicability of one-class classifiers under cases of extreme imbalance, we conduct two sets of experiments. In the first set, we simply compare their performance to multi-class classifiers, in order to establish a baseline performance. In the next set, we employ imbalance rectifying methods as outlined in the previous chapter to make the data sets more conducive for multi-class classifiers. The purpose of these sets of experiments is to understand the levels of imbalance at which such method fail. The results are presented

We use the and the for one-class classification. The binary classifiers used are:

- Naïve Bayes (NB)
- Nearest Neighbour (IBK)
- Decision Trees (J48)
- Multilayer Perception (MLP)
- Support Vector Machines (SVM)

The one-class classifiers employed are:

- Autoassociator (AA)
- One-class support vector machine (ocSVM)

With respect to the sampling methods, we employ:

- SMOTE [14]
- Random oversampling
- Random undersampling
- One-sided selection with Tomek links [50] (one-sided selection is a well known sampling method for handling overlap and noise).

We chose the ocSVM and AA as they are robust classifiers that are able to handle wide variety of complex domains. The ocSVM has been used for, among other things, image retrieval (Chen *et. al.* [16]), document classification (Manevtiz *et. al.* [58]), yeast regulation prediction (Kowalczyk *et. al.* [46]) and payload anomaly detection (Perdisci *et. al.* [63]). Autoencoders have similarly been used in a variety of domains, including tramp metal detection (Bulitko *et. al.*, [12]), intrusion detection (Govolko *et. al.* [28]), HIV classificaiton (Betechoh *et. al.* [9]) and speech emotion recognition (Deng *et. al.* [17]).

For these reasons, we chose to employ them for our experiments. Apart from AA, ocSVM and the sampling methods, all classifiers have been implemented in WEKA [29] and run with their default settings. This is done so as to prevent any bias resulting from the fine tuning the parameters in order to obtain optimal results from specific datasets. The experiments with AA were implemented using the AMORE[1] R package, and run in R[2]. One hidden layer was used for the AA in all the experiments and the number of training iterations was set to 50. The momentum value was set to 0.99, and the learning rate to 0.01. The number of hidden units for the artificial datasets were set to 4. For all other datasets, they varied from 1 to the number of dimensions of the particular dataset, and the number of units giving the best results were chosen. The R implementation of ocSVM via the e1071 package[3] is used. Finally, for sampling, all the methods are implemented in R with the Unbalanced[4] package.

The performance measure we use is the geometric mean of the per-class accuracies [50]. It is given by $g{-}mean = \sqrt{acc_1 \times acc_2}$, where acc_i is the accuracy of the classifier on instances belonging to class i. Note that the metric is computed in a manner that is independent of imbalance, since each class is treated individually. Thus, it is immune to imbalance; it is for these reasons that we selected it for our experiments. Evaluation is done using 5x2 cross validation; folds are generated for both classes for binary classifiers and only for the target class for the one-class classifiers. A dedicated

[1]AMORE: A MORE flexible neural network package, `http://cran.r-project.org/web/packages/AMORE/index.html`

[2]The R Project for Statistical Computing, `http://www.r-project.org/`

[3]`http://cran.r-project.org/web/packages/e1071/index.html`

[4]`http://cran.r-project.org/web/packages/unbalanced/index.html`

outlier set is maintained for binary classifiers for testing purposes, as due to the nature of reducing the size, at the most extreme levels of imbalance there will be hardly any data for thorough evaluation.

To simulate the effect of imbalance, we fix the size of the target set, and exponentially decrease the size of the outlier set, until there are only 4 instances in the outlier set; we generate 20 such outlier sets. Let $o = |outliers|$ be the number of outliers. Then, the size of each outlier set is 4^w, where $w \in S$, an 20-element vector of equally spaced values that range from $width$ to 1. The value of $width$ is calculated as $\frac{\lg o}{\lg 4}$.

We present the results of our experiments in the following section.

3.4 Results

The results in this section are organized by dataset. Each subsection presents the plots showing the trends in performance over the 20 variations of the outliers without sampling, as well as the plots for the best performing sampling method; this is determined as the sampling method that produces the highest result for the smallest outlier dataset (over all classifiers). The remainder of the sampling method plots are displayed in Appendix B.

It is worth noting that in each of the plots, the results for the one-class classifiers appear as horizontal lines. This is because the x-axis represents the level of imbalance, and the performance of one-class classifiers is not impacted by this as they only train on a single class. Thus, the training set for them is the same, irrespective of the level of imbalance.

3.4.1 Results on *Artificial Unimodal* dataset

The trends for no sampling are shown in Figure 3.13, whereas Figure 3.14 shows the trends for classifier performance under undersampling.

As it represents the simplest distribution considered in this chapter, it is unsurprising that the performance is very good for both binary and one-class classifiers. It is only at higher levels of imbalance that the performance starts to deteriorate.

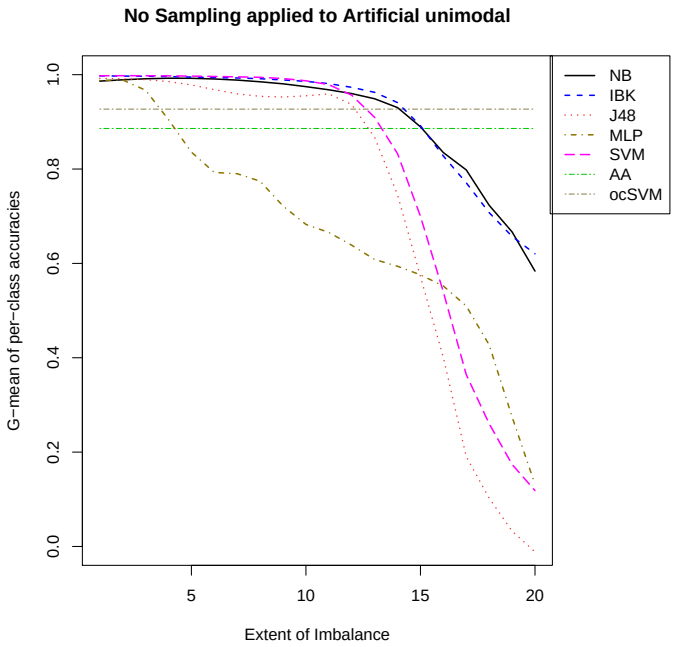


Figure 3.13: The performance trends of various classifiers over the artificial unimodal dataset with no sampling.

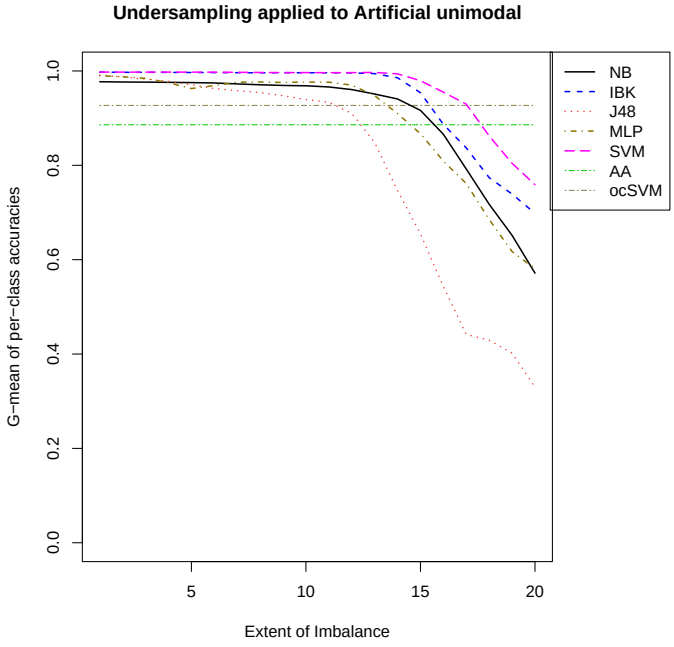


Figure 3.14: The performance trends of various classifiers over the artificial unimodal dataset with undersampling.

3.4.2 Results on *Artificial Multimodal* datasets

The trends for no sampling and applying SMOTE for the more complex artificial multimodal dataset with no overlap are displayed in Figure 3.15 and 3.16, respectively. Figures 3.17 and 3.18 display the results over the dataset with overlap.

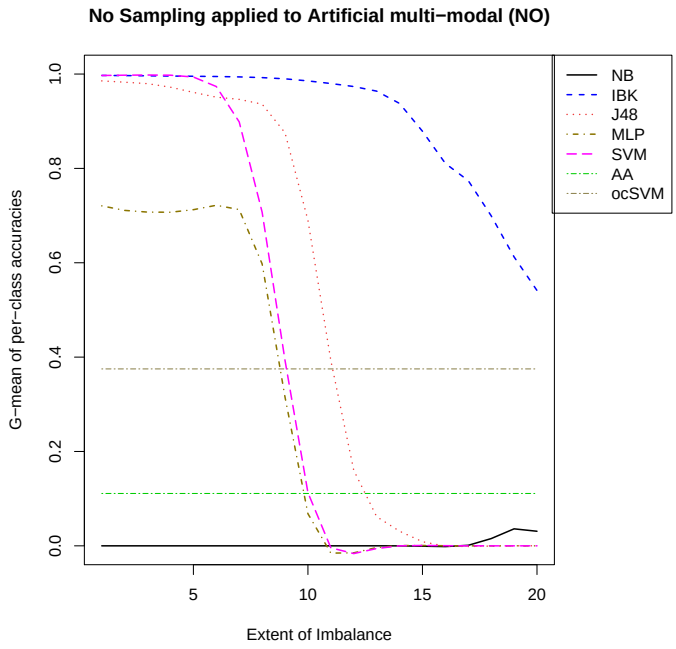


Figure 3.15: The performance trends of various classifiers over the artificial multimodal dataset (no overlap) with no sampling.

Given that these domains are inherently very complex, it is no surprise that the classifier performance deteriorates a lot faster as opposed to the simpler unimodal domain. However, upon applying SMOTE, the performance does stabilize till medium levels of imbalance, but does indeed deteriorate at higher levels.

The more interesting result, perhaps, is the performance of one-class classifiers. Due to the highly complex nature of the domains, they show extremely poor performance. Even at the greatest level of imbalance, they are out-performed by IBk without SMOTE, and by most classifiers after applying SMOTE. In other words, over highly complex domains, the advantage offered by one-class classifiers in extreme imbalance disappears; this is, indeed, the issue this thesis proposes to resolve.

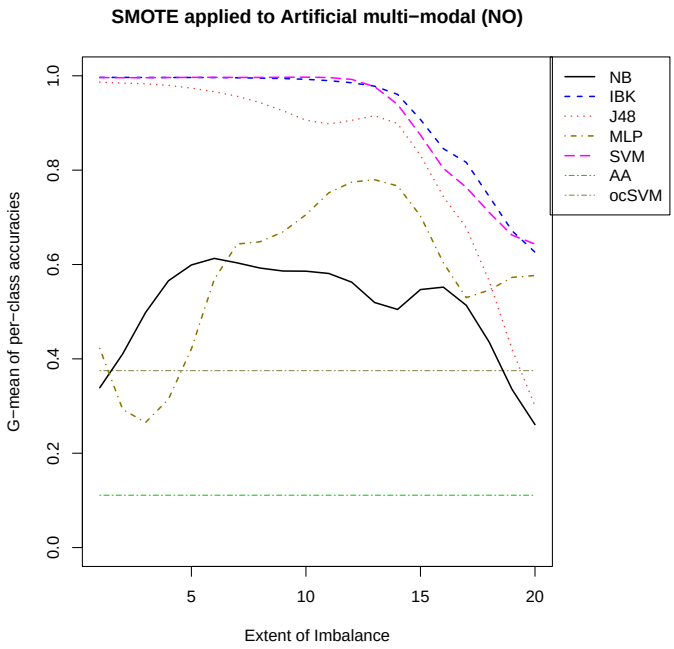


Figure 3.16: The performance trends of various classifiers over the artificial multimoda dataset (no overlap) with SMOTE.

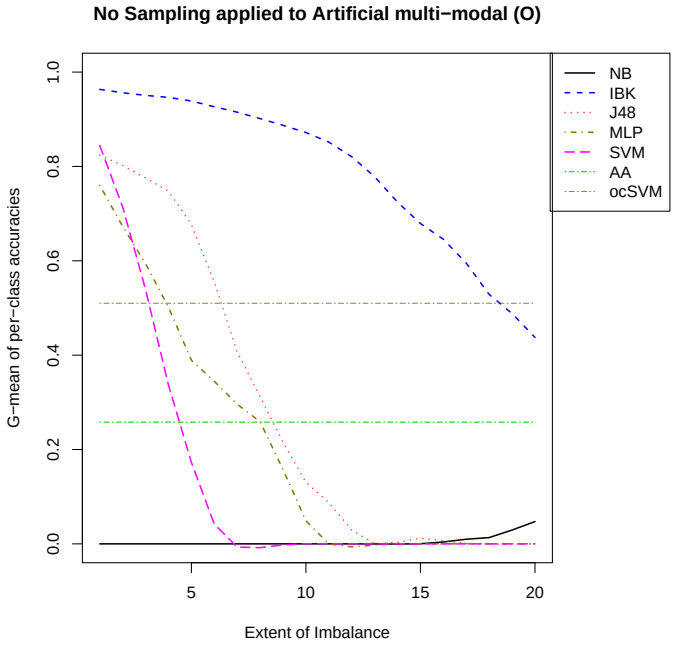


Figure 3.17: The performance trends of various classifiers over the artificial multimodal dataset (with overlap) with no sampling.

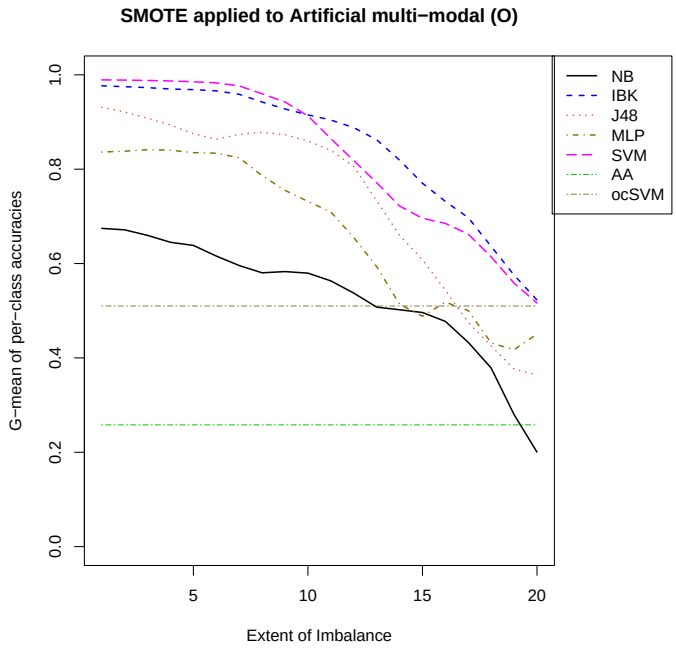


Figure 3.18: The performance trends of various classifiers over the artificial multimoda dataset (with overlap) with SMOTE.

3.4.3 Results on *Diabetes* dataset

Trends of the classifiers without sampling for the Diabetes dataset are displayed in Figure 3.19, and with sampling in Figure 3.20.

This is a complex domain due to a significant overlap between the classes, as outlined in the previous section. This is evident by the performance of all classifiers at the beginning of the trends. Without sampling, all classifiers have a sharp decline in performance; however, undersampling induces a high level of stability in performance. With respect to the one-class classifiers, we observe that without sampling they offer superiority over binary classifiers early on. Their performance, though, is lower than most classifiers augmented with undersampling. Even when looked at on their own, the one-class classifiers do not perform very well. This is likely due to two facts. Firstly, there are only 500 target instances, and during 5×2 cross-validation, only 250 instances are available for training. Secondly, there is a high degree of overlap between the classes. This point may also contribute to the later; more data might be needed to induce better performance.

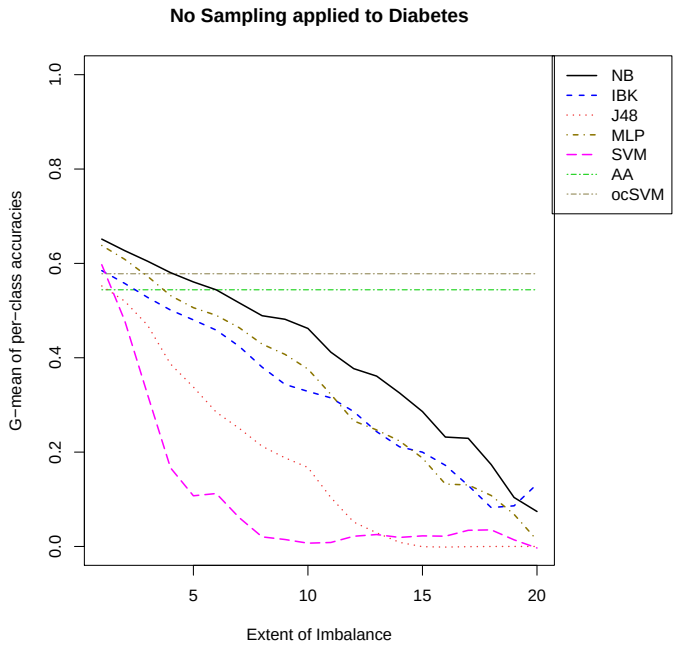


Figure 3.19: The performance trends of various classifiers over the diabetes dataset with no sampling.

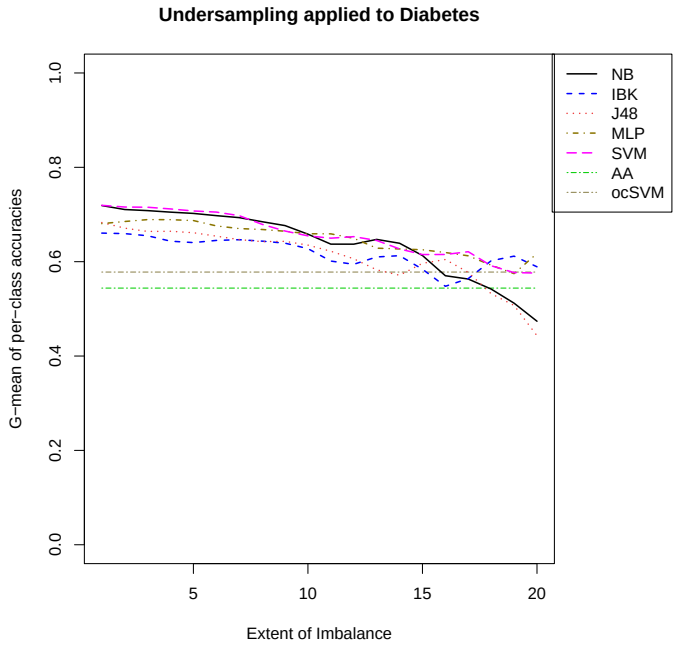


Figure 3.20: The performance trends of various classifiers over the diabetes dataset with undersampling.

3.4.4 Results on *Heart Disease* dataset

Figure 3.21 displays the trends of the classifiers over the heart disease domain with no sampling; the trends with sampling are shown in Figure 3.22.

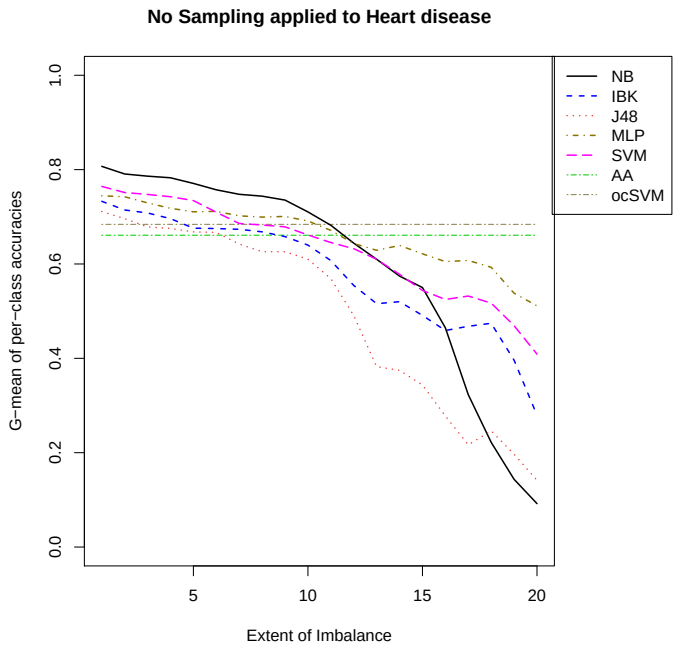


Figure 3.21: The performance trends of various classifiers over the heart disease dataset with no sampling.

As indicated in the previous section, while this dataset is highly multi-modal, with small compact clusters, there is a moderate level of separability between the two classes. Thus, the classifiers start out with relatively high performance. As the imbalance increases, there is a gradual decline in performance in the absence of sampling. With undersampling, the performance stabilizes, though it still shows a decreasing trend. The one-class classifiers perform better as compared to the diabetes dataset. Even when undersampling is applied, ocSVM shows a slight edge in the most extreme case of imbalance. Naturally, as can be seen, this edge is too close for comfort; the relative complexity of this domain takes away the edge they would normally offer in such extreme cases of imbalance.

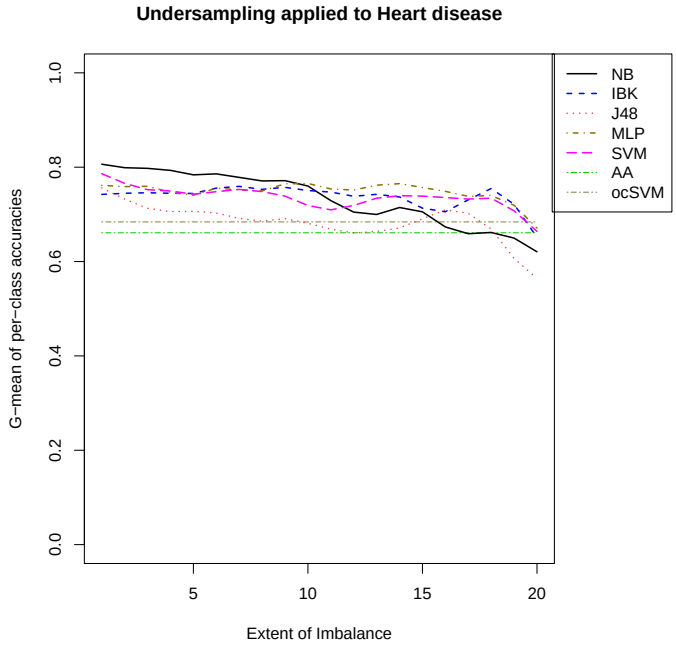


Figure 3.22: The performance trends of various classifiers over the heart disease dataset with undersampling.

3.4.5 Results on *Ionosphere* dataset

Trends for classifiers with no sampling for the Ionosphere dataset are displayed in Figure 3.23. Figure 3.24 shows the trends when undersampling is employed.

This domain, while spread, does not have a high level of overlap between the classes. As a result, all classifiers show high performance. The trends as imbalance increases, as expected, are in a downward direction. Given the relative *easy* nature of the domain, undersampling is able to maintain a respectable performance in extreme cases of imbalance; it is, however, still outperformed by both one-class classifiers considered in this chapter.

3.4.6 Results on *Thyroid Disease* dataset

The trends for classifier performance without and with sampling for thyroid disease dataset are shown in Figures 3.25 and 3.26, respectively.

This is an interesting domain as much of the outlier data is “overshadowed” by

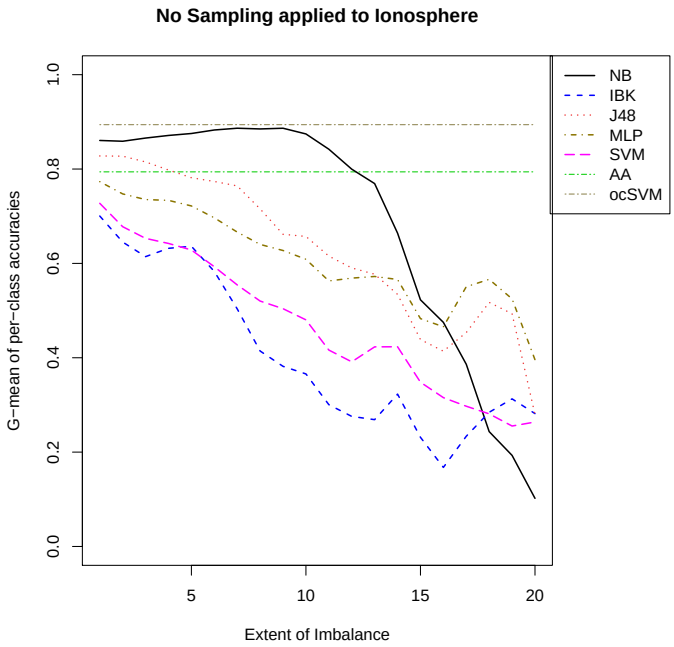


Figure 3.23: The performance trends of various classifiers over the ionosphere dataset with no sampling.

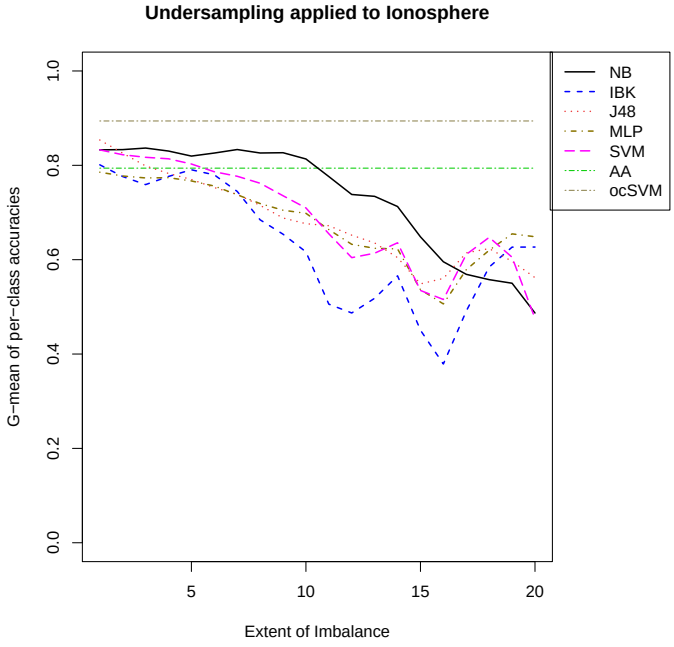


Figure 3.24: The performance trends of various classifiers over the ionosphere dataset with undersampling.

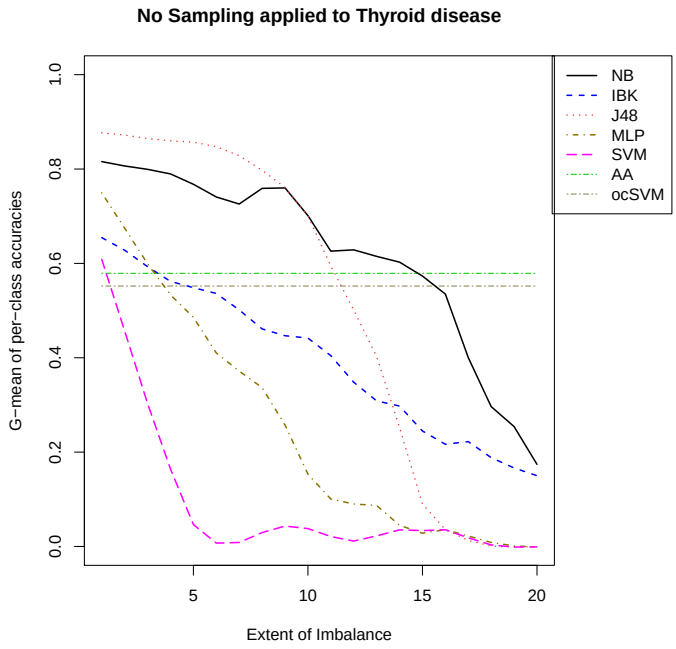


Figure 3.25: The performance trends of various classifiers over the thyroid disease dataset with no sampling.

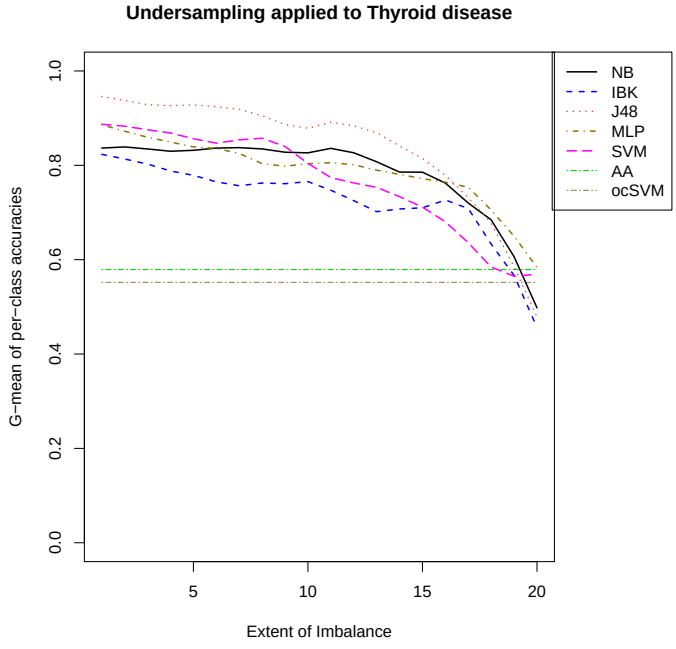


Figure 3.26: The performance trends of various classifiers over the thyroid disease dataset with undersampling.

the target data (as discussed in the previous section). Binary classifiers are able to perform relatively well as the target class, while very spread, is easy discriminated with the outlier class. This is evident by their performance at low levels of imbalance. As it increases, though, the performance declines sharply without sampling. Undersampling is able to reduce the rate of decline, but at high levels the performance becomes weak. One-class classifiers, given the nature of the target data and the massive overlap, perform very poorly. Even at the most extreme levels of imbalance, some classifiers when augmented with undersampling are still slightly better than both one-class classifiers.

3.4.7 Results on *Sonar* dataset

Classifier trends without sampling for the sonar dataset are shown in Figure 3.27, and with sampling in Figure 3.28.

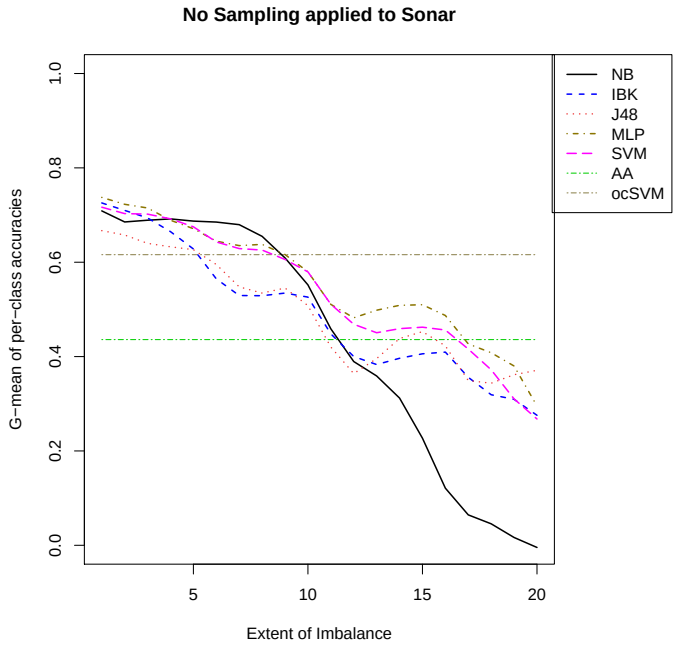


Figure 3.27: The performance trends of various classifiers over the sonar dataset with no sampling.

The dataset shows significant overlap in some dimensions. However, it's biggest drawback is the combination of high dimensionality and the general lack of data from

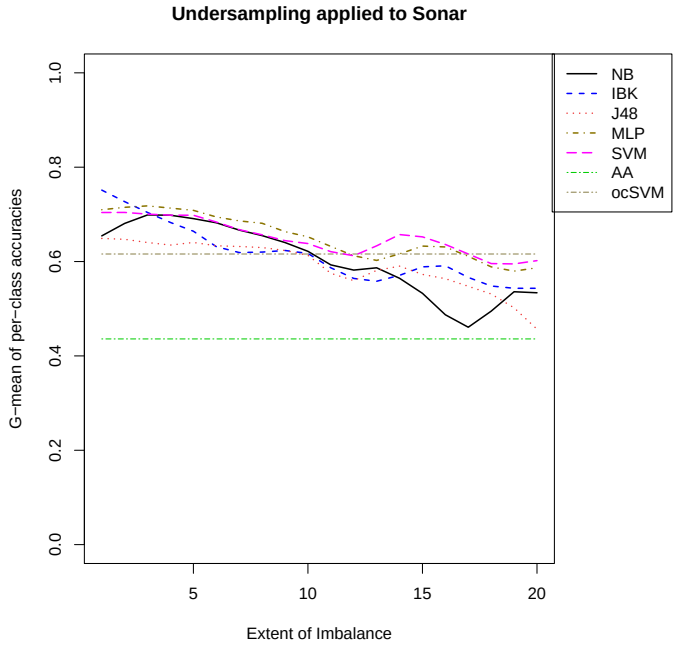


Figure 3.28: The performance trends of various classifiers over the sonar dataset with undersampling.

both classes; there are only 111 target instances overall, and 48 outlier instances, whereas there are 60 dimensions. This naturally impacts the performance of both binary and one-class classifiers. For the binary classifiers, there is a steady decline in performance as imbalance increases. With sampling, while there is still a decline in performance, the rate is less severe. Amongst the one-class classifiers, the AA performs very poorly, whereas the ocSVM shows better performance. Indeed, it is better than all the binary classifiers at the most extreme levels of imbalance. This improvement, however, is not as large as one would expect, or at the very least hope. This can well be attributed to the lack of data, the high dimensionality and the overlap between the two classes.

3.4.8 Results on *Alphabets* dataset

Figures 3.29 and 3.30 display the trends for various classifiers over the alphabets dataset without and with sampling, respectively.

This domain has a significant overlap between the target and outlier class, along

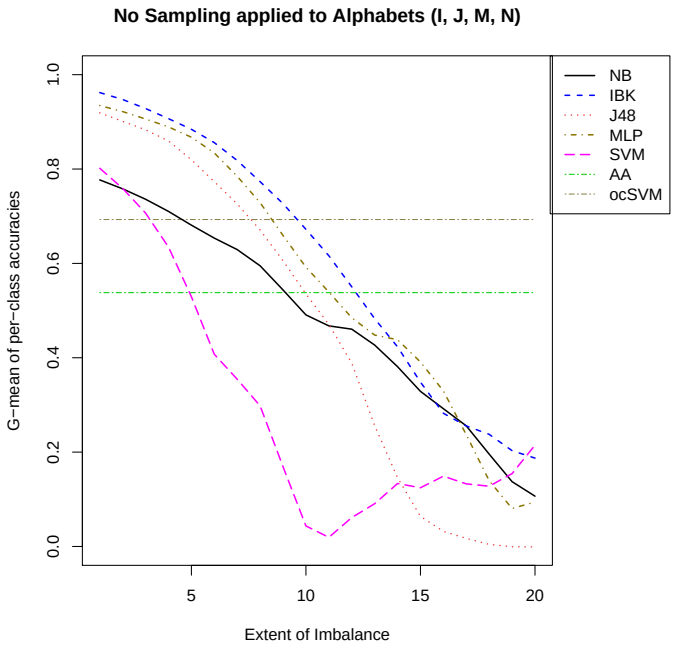


Figure 3.29: The performance trends of various classifiers over the alphabets dataset with no sampling.

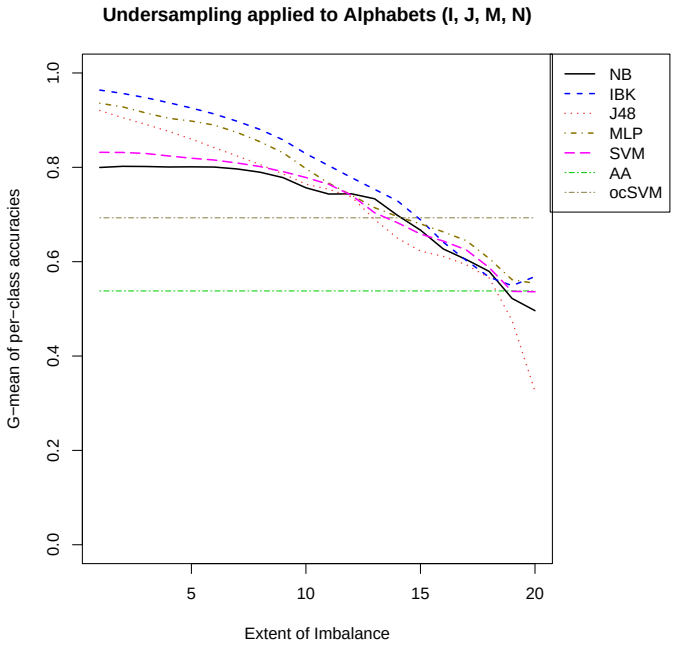


Figure 3.30: The performance trends of various classifiers over the alphabets dataset with undersampling.

with a multi-modal distribution. The bulk of the data, however, is consolidated in one mode. Moreover, there is ample data, and the domain has a relatively low dimensionality. These factors allow binary classifiers to have relatively high performance. An increase in imbalance, though, causes a severe decline in performance; even with undersampling we observe a significant decline in performance. With respect to the one-class classifiers, while the AA has low performance, the ocSVM has a relatively higher performance; it outperforms all classifiers complimented with undersampling. Due to the complexity of the dataset, however, even though there is a large amount of data to learn from, the one-class classifiers do not perform as well as we would expect.

3.4.9 Results on *Forest* dataset

The trends for various classifiers on the forest dataset without sampling are displayed in Figure 3.31, and with sampling with Figure 3.32.

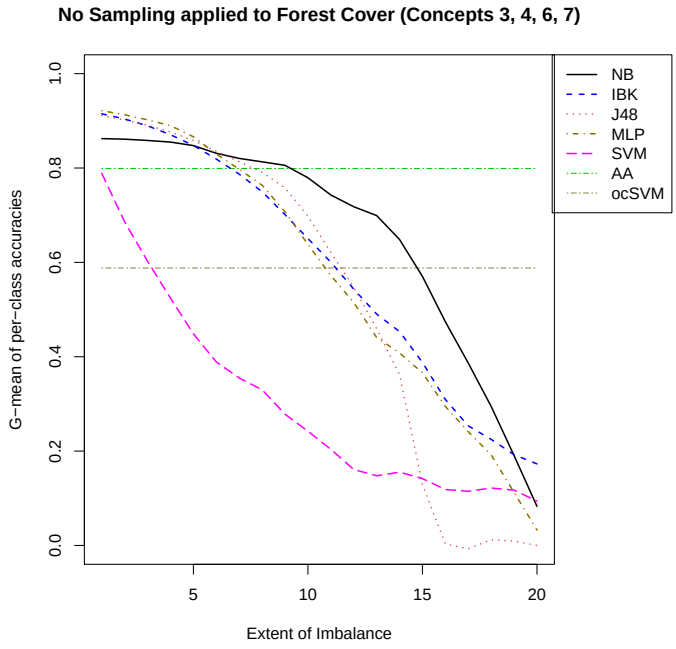


Figure 3.31: The performance trends of various classifiers over the forest dataset with no sampling.

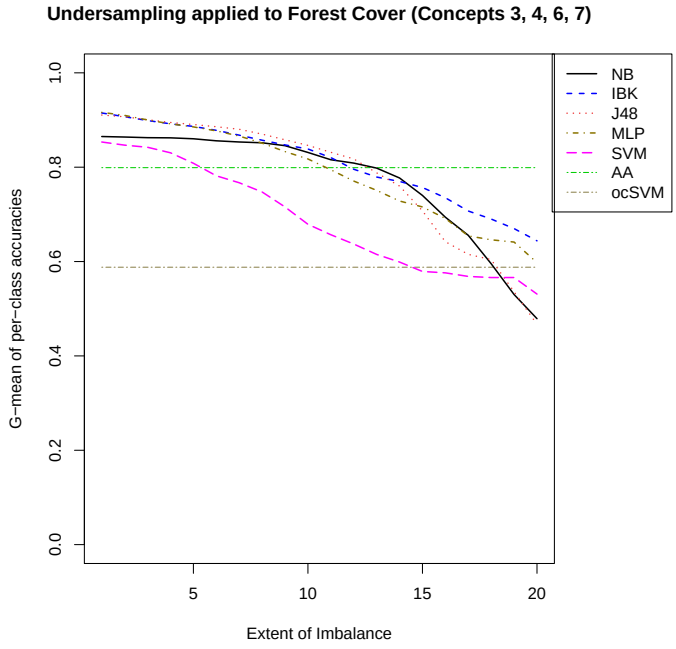


Figure 3.32: The performance trends of various classifiers over the forest dataset with undersampling.

There is significant overlap between the certain dimensions in the domain. Specifically, if we observe the plots presented in the previous section, in the first two components, the target class is almost completely overwhelmed with the outlier class. However, in when observed from the third component (especially with the first component), we observe that the target class spreads "outside" of the outlier class. Thus, the target class is distinguishable from the outliers in a small subspace. This is observed in the high performance of the binary classifiers, as well as the AA. While the performance degrades sharply as the rates of imbalance increase, undersampling is able to slow down the decline. However, at high levels of imbalance, the AA outperforms all classifiers rectified by undersampling by a considerable margin. The ocVSM, on the other hand, performs rather poorly.

3.4.10 Results on *ForestC1* dataset

The trends for classifiers without sampling over the forestC1 dataset are shown in Figure 3.33, and in Figure 3.34 for sampling.

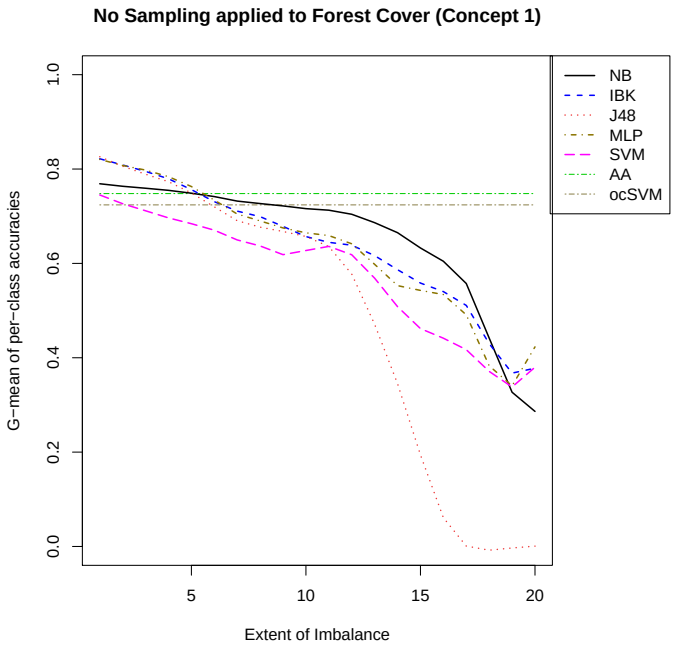


Figure 3.33: The performance trends of various classifiers over the forestC1 dataset with no sampling.

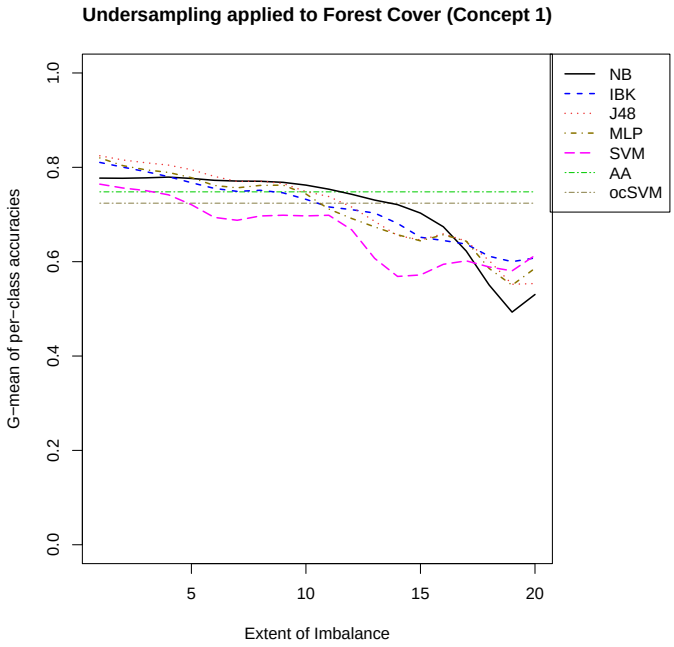


Figure 3.34: The performance trends of various classifiers over the forestC1 dataset with undersampling.

This domain is a rather simple, unimodal domain. However, there is still a significant overlap between the two classes. While all binary classifiers start out with high performance, there is a slow by steady decline in performance as imbalance increases. This decline is, once again, slowed by undersampling. At high levels of imbalance, all classifiers, with and without undersampling, are outperformed by both one-class classifiers.

3.4.11 Results on *ForestC2C5* dataset

Figure 3.35 shows the classifier trends over forestC2C5 without sampling, whereas Figure 3.36 shows the trends with sampling.

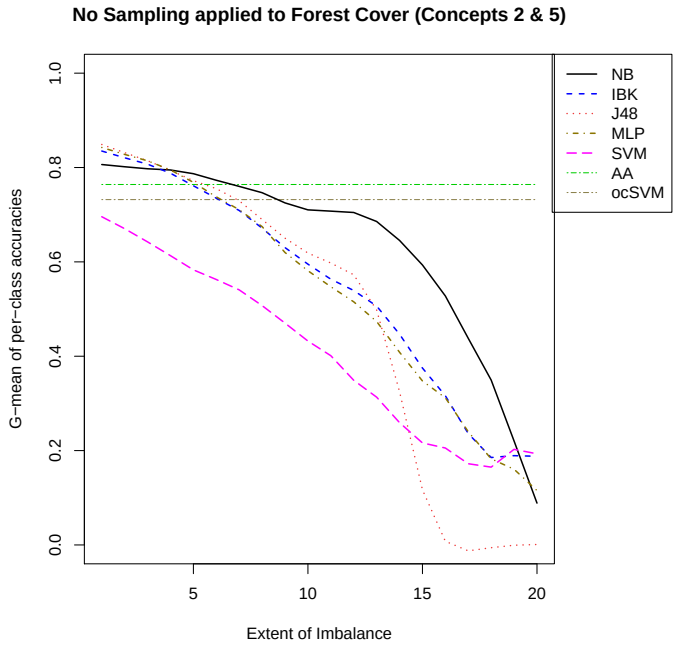


Figure 3.35: The performance trends of various classifiers over the forestC2C5 dataset with no sampling.

While there are indeed two different tree species within the target class, they are highly similar to each other, and thus the resulting distribution is unimodal. When compared to forestC1, however, the level of overlap is much less. The trends in performance of binary classifiers, when no sampling is applied, indicate a steeper rate in decline. This rate is less severe when oversampling is applied. As with forestC1,

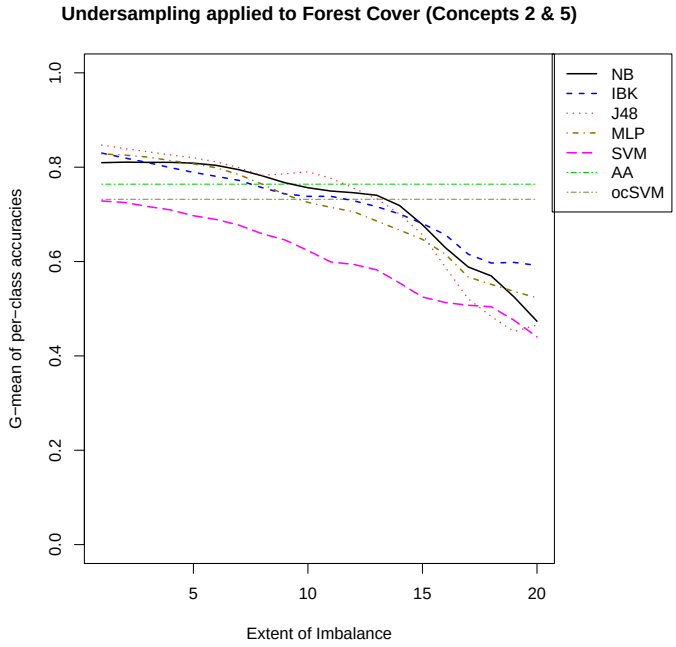


Figure 3.36: The performance trends of various classifiers over the forestC2C5 dataset with undersampling.

all binary classifiers are outperformed by the two one-class classifiers, even when undersampling is considered.

3.5 Discussion

The experiments conducted in this chapter aimed to investigate the performance of binary classifiers as imbalance increases and the effectiveness of various sampling methods in maintaining good classifier performance in the most extreme cases of imbalance. In the absense of sampling, all binary classifiers exhibit a sharp decline in performance; the rate at which the decline happens is naturally dependent upon the complexity of the data, the dimensionality, and the number of instances available for induction. The performance of the one-class classifiers is also affected by these factors. In particular, if we observe the results over the two artificial datasets, the one-class classifiers exhibit very high performance over the simpler dataset but perform very poorly over the highly complex multi-modal dataset. The same goes for the binary

classifiers; the rate of decline is steeper, even with sampling, in the multi-modal case.

With respect to the UCI datasets, about half of all the datasets have a relatively small amount of data available for training. With respect to one-class classifiers, it can be argued that this offers an atypical scenario, as typically there is an overabundance of data from the target class. However, these are included in our experiments for completeness. The diabetes and heart disease datasets, while showing a steep decline in performance without sampling, stabilize relatively well with undersampling, and it produces better performance in the most extreme cases of imbalance over one-class classifiers. A similar observation is made in the thyroid disease dataset; the performance of binary classifiers is less stable than diabetes or heart disease, but in the most extreme case of imbalance, the MLP classifier outperforms both one-class classifiers. The poor performance of one-class classifiers in this dataset, even though there is ample data, can be attributed to the fact that the outlier distribution is almost completley contained within the much more spread target distribution.

The alphabets and forest cover derivative datasets were artificially separated into target and outlier classes so as to induce some level of complexity and maintain some level of "common sense" in the separation. All these domains exhibit a significant level of overlap between the two classes, along with varying levels of complexity. All binary classifiers suffer tremendously as the levels of imbalance increase. Undersampling, however, does significantly improve classifier performance at more extreme levels of imbalance. The alphabets dataset is perhaps the most complex of these domains, and consequently, we observe that, without undersampling, the rate of decline is the greatest, and even with undersampling, at the most extreme level of imbalance, the performance is lower compared to the forest cover derivatives. Even the one-class classifiers perform poorly on the alphabets dataset as compared to the forest cover derivatives. Indeed, in the most extreme case of imbalance, most binary classifiers perform better than the AA, and are better than the ocSVM for all but the most extreme cases of imbalance. Over the forest cover domains, the performance of one-class classifiers is remarkably good. In particular, both the AA and the ocSVM perform equally well over the *forestC1* and *forestC2C5*, domains which are less "complex"

than *forest*. In the later, the AA performs well, though the ocSVM suffers.

3.6 Summary

Given the inherent imbalance present in most real world datasets, it is natural to wonder which classification paradigm would be suitable, i.e., one based on discrimination (binary classification), or one based on recognition (one-class classification). We investigate the performance of binary and one-class classifiers over datasets in which we purposely decrease the size of the outlier (or second) class, thereby increasing the level of imbalance between classes. The results show that in all cases, while the performance of the binary classifiers decreases as the imbalance increases, sampling methods go a long way to mitigate this decline. However, at extreme levels, even sampling does not help. In such a scenario, one-class classifiers become the only viable option.

Unfortunately, we observed that if the domains are complex (specifically in terms of the multimodality and overlap), one-class classifiers are impacted as their performance is lowered. Indeed, in the two artificial multi-modal domains, the diabetes domain, the heart disease domain and the thyroid disease domain, at the most extreme levels of imbalance, the performance of the one-class classifiers is worse than some of the binary classifiers complimented with sampling; in the sonar domain, the performance is just slightly better. Thus, we see that the advantage one-class classifiers offer at extreme levels of imbalance disappears when faced with complexity. In such scenarios, it is natural to wonder whether it is even worth considering their application. In the next chapter, we develop a framework that is aimed at resolving this dilemma. In particular, we will demonstrate that if we force one-class classifiers to learn explicitly along sub-concepts, their performance over complex domains can be improved to levels where they offer the best performance; their poor performance is a direct result of over-generalizing over complex domains, and by learning over sub-concepts, this over-generalization can be prevented.

Chapter 4

Learning the Sub-Conceptual Layer

In the previous chapter, we established that, when there is an extreme level of imbalance in the data, the applicability of binary classifiers becomes futile, even when they are complimented with sampling. One-class classifiers become a promising option; however, as we also observed, if the domain is complex, the advantage that they offer over binary classifiers becomes insignificant. This complexity can arise due to overlap as well as multi-modality, and generalizing over them can lead to poor classifier performance. Therefore, if we wish to improve performance, it is imperative to handle these complexities appropriately. A step towards this end is to understand what causes a domain to exhibit them. We propose that a root cause for these complexities is the presence of multiple sub-concepts that underlie the domain space, an idea we explored in Sharma *et. al* [70]; each of these sub-concepts represent a unique property, or class, within the overall domain, and can typically be identified by an expert in the domain. Thus, by identifying and isolating these concepts and learning over them individually, one can mitigate the effects of domain complexity and induce more accurate one-class classifiers.

We begin the chapter by introducing the framework for one-class classification over sub-concepts in Section 4.1. We then introduce the results of employing this method for classification. The datasets employed are the same as in the previous chapter. The experimental framework employed is described in Section 4.2, and the associated results are presented and discussed in Section 4.3 and Section 4.4,

respectively. Section 4.5 offers a summary of the work presented in this chapter, along with the key insights gained.

4.1 One-Class Classification over Sub-Concepts

The performance of one-class classifiers is severely impacted if the domain under consideration is *complex*. These complexities can arise due to overlap as well as multimodality, and generalizing over them can lead to poor classifier performance. Therefore, if we wish to improve performance, it is imperative to handle these complexities appropriately. A step towards this end is to understand what causes a domain to exhibit them. We propose that a root cause for these complexities is the presence of multiple sub-concepts that underlie the domain space, an idea explored in [70]; each of these sub-concepts represent a unique property within the overall domain, and can typically be identified by an expert in the domain. Thus, we hypothesize that by identifying and isolating these concepts and learning over them individually, one can mitigate the effects of domain complexity and induce more accurate one-class classifiers. In this section, we formally introduce our framework for one-class classification over sub-concepts.

We begin by delving into the notions of *main concepts* and *sub-concepts*. As an example to illustrate the general idea, let us consider a simple learning problem that involves distinguishing between spoiled beans and fresh beans within the family of *common beans* (*Phaseolus vulgaris*). The domain of common beans has a number of different *aspects*, each corresponding to a type of bean. For simplicity, in our example, we only consider three aspects: kidney beans, pinto beans and white beans.

Our learning task is to learn the *concept* of *fresh beans*. Due to the presence of three different aspects, this concept is represented by three *sub-concepts*: *fresh pinto beans*, *fresh kidney beans* and *fresh white beans*.

Figure 4.1 illustrates the generalization our bean example to concepts and sub-concepts. In particular, it illustrates a domain with three aspects and two classes:

- The *target* class is the class over which we will induce a one-class classifier. In

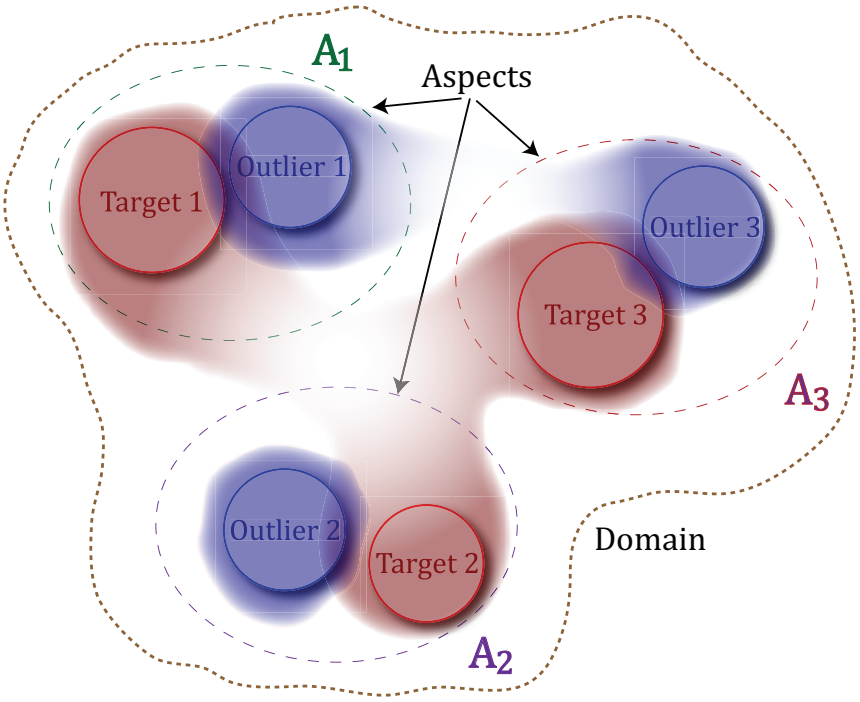


Figure 4.1: The notion of main and sub-concepts.

our bean example, this would represent fresh beans. The *outliers* thus would be spoilt beans.

- The different *aspects* of the domain are denoted by A_1, A_2 and A_3. In our bean example, these would be kidney, pinto and white beans.

- The *concept* which we wish to learn is represented by the target class within each aspect. Thus, the concept is represented by (*Target 1* $\cup$ *Target 2* $\cup$ *Target 3*), and *Target 1*, *Target 2* and *Target 3* correspond to the *sub-concepts* of the concept. For our bean example, *Target 1* would represent fresh kidney beans, *Target 2* fresh pinto beans and *Target 3* fresh white beans, the concept to learn being that of fresh beans.

The gradient in the image represents density; darker regions contain the bulk of the data. The image further serves to illustrate how the presence of multiple aspects can cause the domain to exhibit both *multi-modality* and *overlap* between the classes.

In the presence of sub-concepts, there are two approaches that can be employed for inducing a classifier to learn the main concept:

- Learn the concept as a whole: In this approach, learning is done for the concept over all aspects of the domain. In our bean example, the training data would consist of fresh beans of all types (aspects).

- Learn over sub-concepts: For this approach, learning is done over each sub-concept corresponding to the different aspects of the domain. In our bean example, the training data would be divided based on the type of fresh beans and three classifiers would be built, one for each sub-concept.

Thus, in the first approach, we ignore the presence of sub-concepts and simply learn the entire concept, whereas in the second approach, we tailor our learning methodology to explicitly handle the different sub-concepts. The question thus becomes, which approach to take? Specifically, how does making a conscious decision to learn along sub-concepts impact learning?

To answer this question, let us once again consider the domain illustrated in Figure 4.1. If we are to induce a one-class classifier over the target class without taking each sub-concept into account, we get a classifier as shown in Figure 4.2. Given the complexity of the domain, the classifier would over-generalize over the sub-concepts, the consequence being that while it would indeed cover most of the target class, it would erroneously classify most of the outlier data as belonging to the target class as well.

If we are to acknowledge the presence of aspects and divide the target data based on the associated sub-concepts, on the other hand, we would get a classifier (or a set of classifiers) as shown in Figure 4.3. By learning over the sub-concepts, we do not risk over-generalization; each classifier focuses only on the targets belonging to a single sub-concept. Thus, the resulting classifier would have a much lower error rate over the outliers.

To summarize, we hypothesize that, in the presence of aspects, learning directly over the concept can yield poor performance, especially due to over-generalization.

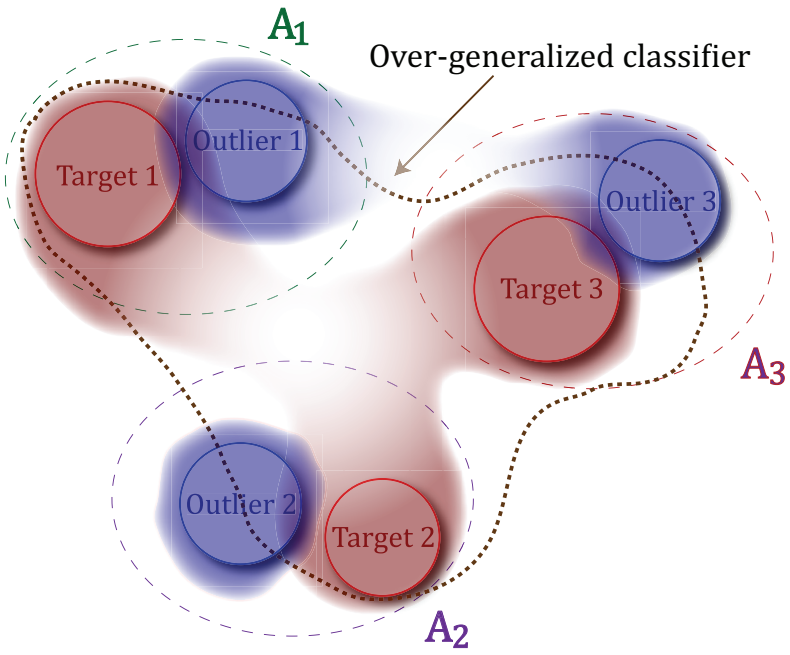


Figure 4.2: Inducing a classifier without distinguishing between different sub-concepts.

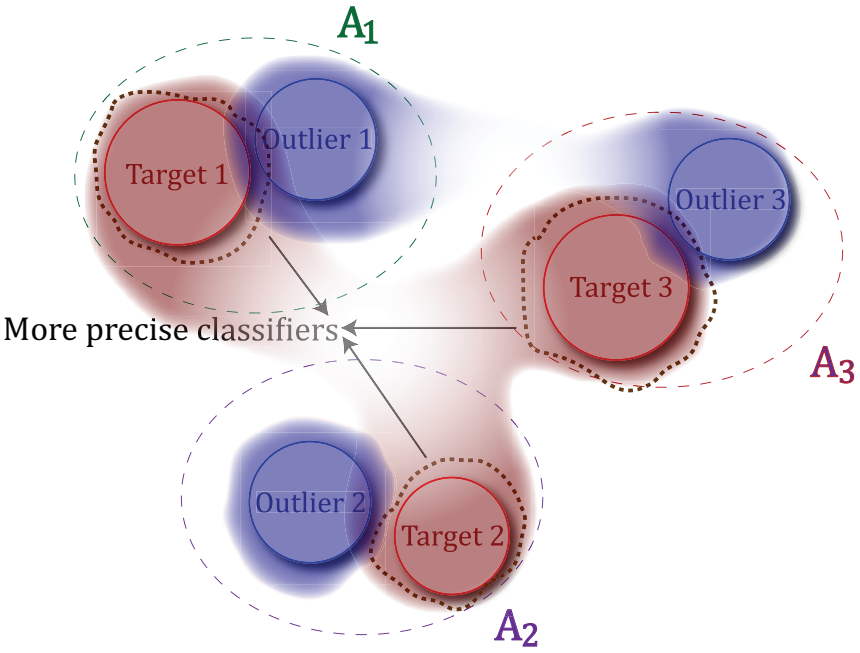


Figure 4.3: Inducing classifiers by distinguishing between different sub-concepts.

Using domain knowledge to identify the aspects, and consequently the sub-concepts, we can prevent a classifier from over-generalizing, thus resulting in better classification. In subsequent sections, we will empirically validate this observation. The question now is, how to actually identify the aspects? Depending on the nature of the aspects and the associated domain knowledge, we propose three different frameworks. These are elaborated in the following sections.

4.1.1 One-Class Classification with Complete Knowledge

The most ideal case occurs when it is possible to identify, with full confidence and accuracy, which aspect a novel instance from the data space belongs to. With respect to training, a domain expert is available to identify the different aspects, and categorize the training data into appropriate sub-concepts, and we can build one-class classifiers over these sub-concepts. Furthermore, once we have built an ensemble of one-class classifiers, the system knows to which aspect a novel instance belongs. Thus, novel instances can be processed by the one-class classifier tailor-made for that particular aspect.

This idea is illustrated in Figure 4.4. Consider a domain in which there are two aspects, **A1** and **A2**. In practice, the domain expert will provide us with training data from both aspects, from which we induce two one-class classifiers **OC 1** and **OC 2** using target data **T1** corresponding to **A1** and **T2** corresponding to **A2**. Now, with respect to classification, for every novel instance i, we are able to identify which aspect it belongs to. If i belongs to **A1**, we classify it using classifier **OC 1**, and by **OC 2** if it belongs to **A2**.

To summarize, one can employ this framework if:

- We have knowledge of which aspect the training samples belong to.
- We are able to identify which aspect novel testing samples belong to.

In order to illustrate the applicability of the framework, let us consider our bean example from earlier. As the *aspects* (*i.e.* the type of beans) are fully identifiable

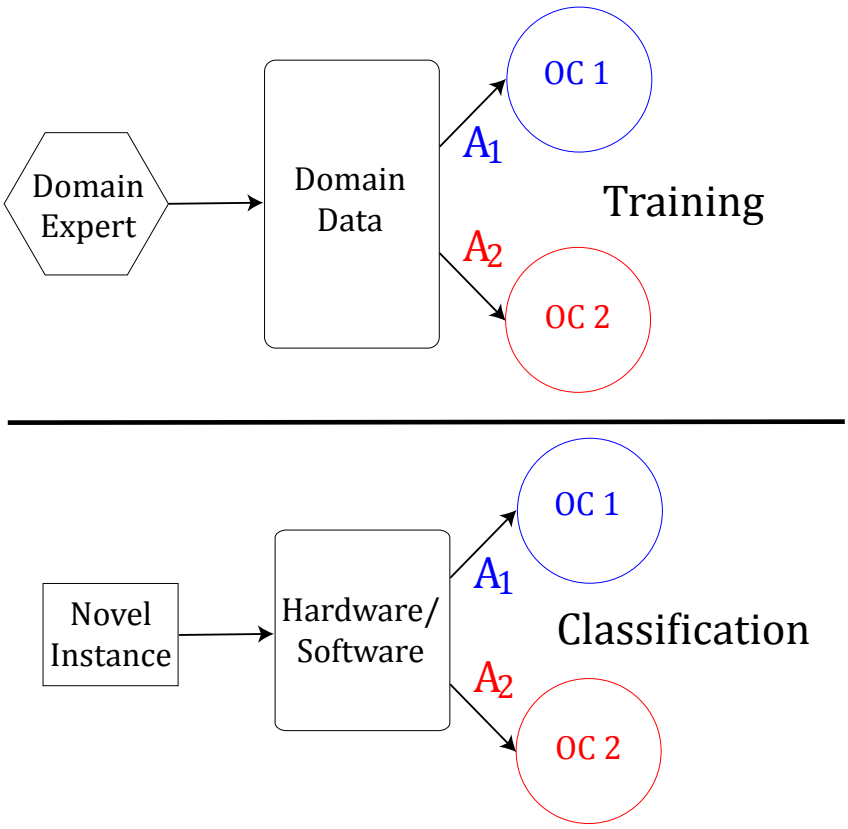


Figure 4.4: One-class classification with full knowledge

in this framework, for training, we would know which *sub-concept* the training data belongs to, and so we would have three training sets for each target sub-concept, one each for fresh kidney, pinto and white beans. Thus, we would end up with three one-class classifiers, each representing a single type of fresh bean. During classification, we would have a mechanism that would be able to tell whether the bean to be classified is a kidney bean, pinto bean or white bean. The key point to note here is that the mechanism would simply detect the type of bean, and not whether it is fresh or spoilt; that is the task of the one-class classifiers. In other words, we are identifying the underlying aspects of the domain. Based on the type, the bean would be sent to the appropriate one-class classifier that would then decide whether it is fresh or spoilt.

4.1.2 One-Class Classification with Fuzzy Knowledge

In reality, it may or may not be possible to identify the underlying aspects in the domain. This may occur either due to the inability of the available hardware or software to perform such a task without human intervention, or because the underlying processes that conform to the aspects themselves are very difficult to quantify; the concepts are *fuzzy*.

For the case of fuzzy knowledge, we propose the following framework: training the one-class classifiers is done as in the framework for full knowledge by employing the target data. However, since the hardware cannot explicitly identify the concept, we induce a multi-class classifier over the known aspects. Depending on how the aspects are represented in the domain, the data employed for this may or may not include outlier instances; this is because we are learning to differentiate between the aspects, and not targets and outliers. Now, when novel instances are encountered, they are classified into the appropriate aspect by the multi-class classifier, and processed by the appropriate one-class classifier. The framework is outlined in Figure 4.5.

To summarize, one can employ this framework if:

- We have knowledge of which aspect the training samples belong to.

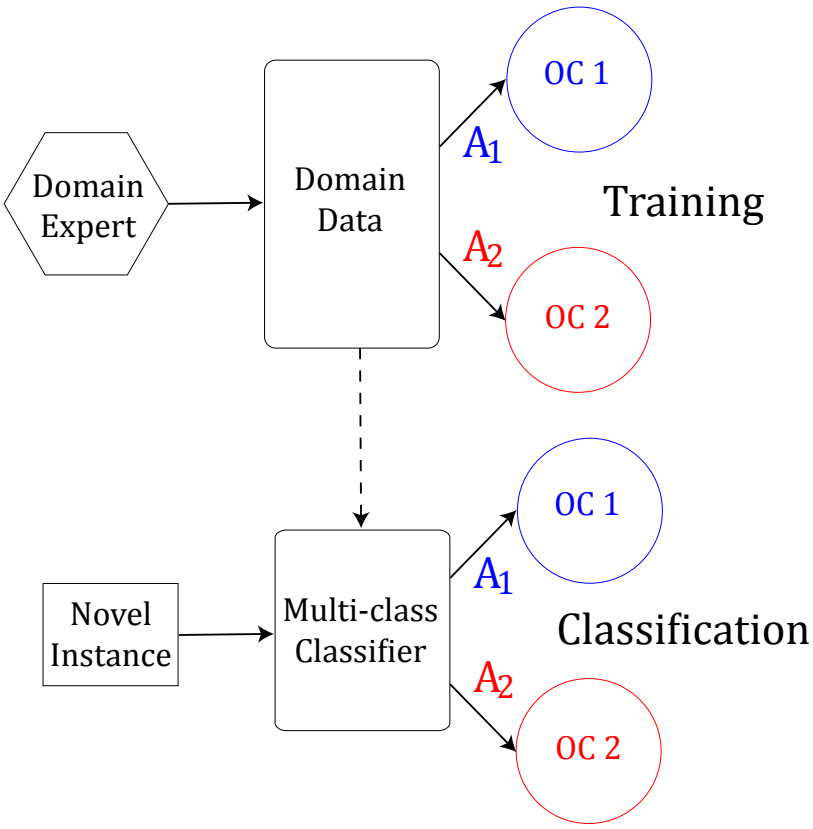


Figure 4.5: One-class classification with fuzzy knowledge.

- We are unable to identify which aspect novel testing samples belong to.

Let us now consider how this framework would be applied to our bean example. The *aspects* (*i.e.* the type of beans) in this case would be identifiable during training, but during classification, we would have no way of knowing whether the bean to be classified for freshness is a kidney, pinto or white bean. Thus, we would first employ supervised learning for learning the underlying aspects of the domain in order to aid us during the classification phase. Specifically, we would have a training set composed of kidney, pinto and white beans, irrespective of whether they are fresh or spoilt, and we would train a multi-class classifier over the type of beans. For training the one-class classifiers over the target concept, we would have three training sets, one for each *sub-concept*, corresponding to fresh kidney, pinto and white beans. Thus, the multi-class classifier learns the aspects of the domain, and the one-class classifiers learn the target sub-concept corresponding to each aspect. During classification, a bean would be first passed to the multi-class classifier in order to identify its type, and then it would be passed to the appropriate one-class classifier to ascertain whether it is fresh or spoilt.

4.1.3 One-Class Classification with No Knowledge

The worst case scenario occurs when we have no knowledge of the aspects underlying the domain, for example, due to the lack of a domain expert in the field. In order to divide the domain into the unknown aspects, we turn to classical unsupervised learning methods: clustering. The purpose of clustering is to divide the data space into a number of regions such that instances in a particular region are most similar to each other; this is illustrated in Figure 4.6. Naturally, instances that are affected by the same aspect will be most similar to each other, and thus, in the absence of knowledge of aspect, we simply cluster the data space and build one-class classifiers over each cluster, the aim being that each cluster will represent an unknown aspect. The final classifier is an ensemble of all the various classifiers built on the clusters. Classification is done as follows: If an instance is positively classified by at least one

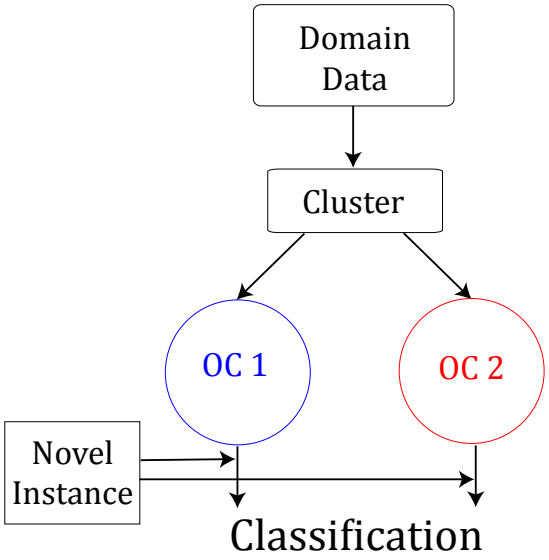


Figure 4.6: One-class classification with no knowledge.

of the models, then it is assigned to the *target* class; otherwise, it is classified as an *outlier*.

Thus, this framework is to be applied when:

- We have no knowledge regarding what aspects occur in the domain.

This framework would be applied to our bean example if we have absolutely no idea as to what type of beans will be given to us. In other words, we have no knowledge regarding the *aspects* of the domain. We would simply cluster the training data of fresh beans, and induce one-class classifiers over each cluster. During classification, each bean would be passed to these classifiers, and would be classified as fresh if one of the classifiers deems it to be so.

In the following section, we conduct a mathematical analysis of the frameworks described in this section. Given that the clustering framework represents the worst case scenario, we analyze with respect to this framework.

4.1.4 One-Class Classification along Sub-Concepts: A Mathematical Perspective

The analysis is given in the context of *posterior densities*, *i.e.*, by viewing one-class classification as estimating the density function of the distribution of instances from the given class. It should be noted that the word *distribution* has a dual meaning; in this context, it can refer to either the actual posterior density of the data, or a *region of similarity*, in which each instance is "similar" to all other instances in the cluster, where similarity is defined by the nature of clustering. We use the term *distribution* interchangeably, either in a probabilistic context, or a context based on the idea of similarity.

Let X represent the set of instances under consideration, and ω be the class to which they belong. What we are interested in obtaining is $P(X|\omega)$, the actual posterior probability density function (pdf). Knowing this can allow for one-class classification by imposing a threshold τ on the value of this function, *i.e.*,

$$\text{Classification}(x \in X) = \begin{cases} \text{target}, & \text{if } P(x \in X|\omega) \geq \tau \\ \text{outlier}, & \text{otherwise} \end{cases} \tag{4.1}$$

However, in practice, the best we can do is obtain an estimate $\hat{P}(X|\omega)$ of $P(X|\omega)$. Given this estimate, and the classifier formulation given in Eq. (4.1), there are two sources of error that can occur when using $\hat{P}(X|\omega)$:

ϵ_t: The probability that we classify a target instance as an outlier instance (a false negative).

ϵ_o: The probability that we classify an outlier instance as a target instance (a false positive).

We divide X to obtain c sub-concept clusters X_i, where $X = \bigcup_{i=1}^{c} X_i$. The clusters may or may not be disjoint. We treat each cluster i as belonging its own unique class ω_i, having its unique pdf $P(X_i|\omega_i)$. Performing one-class classification on these clusters is equivalent to obtaining an estimate $\hat{P}(X_i|\omega_i)$ of $P(X_i|\omega_i)$. As

before, each $\hat{P}(X_i|\omega_i)$ will have its own two sources of error, namely ϵ_t^i and ϵ_o^i. Let ϵ_t^M and ϵ_o^M denote the error of the combined model, composed of the various models built over the clusters.

Since each cluster represents a simpler distribution as compared to the original distribution, a one-class learner should be able to model a cluster more efficiently than the original distribution[1]. In other words, $\forall i \in [1, c], (\epsilon_t^i \leq \epsilon_t) \wedge (\epsilon_o^i \leq \epsilon_o)$, and consequently, $\epsilon_t^M \leq \epsilon_t$ and $\epsilon_o^M \leq \epsilon_o$.

We will now attempt to derive a relationship between the combined model errors, ϵ_t^M and ϵ_o^M, and the error of the single model over X, ϵ_t and ϵ_o, for both cases of error, using the assumption stated in the previous paragraph.

- *Error of False Negatives:* For the combined model, this will occur when a target instance is rejected by all of the cluster models. Since the probability of a single cluster model i rejecting a target instance is ϵ_t^i, and each ϵ_t^i is a mutually independent event, the probability of the combined model rejecting a target instance is $\prod_{i=1}^{c} \epsilon_t^i$. Based on the aforementioned hypothesis, since $\epsilon_t^i \leq \epsilon_t$, we have $\prod_{i=1}^{c} \epsilon_t^i = \epsilon_t^M \leq \epsilon_t$.

- *Error of False Positives:* For the combined model, this will occur when an outlier instance is incorrectly accepted by any one of the cluster models. Since the probability of a single cluster model i accepting an outlier is ϵ_o^i, and each ϵ_o^i is a mutually independent event, the probability of the combined model accepting an outlier instance is $\sum_{i=1}^{c} \epsilon_o^i$. In order for ϵ_o^M to be less than or equal to ϵ_o, in the simplest case, if we assume all ϵ_o^i to be equal, a necessary condition is that each $\epsilon_o^i \leq \frac{\epsilon_o}{c}$. However, given that the distribution of instances represented by the clusters is far simpler than the original, more complex distribution, we assume that all ϵ_o^i will have values to ensure that $\sum_{i=1}^{c} \epsilon_o^i \leq \epsilon_o$.

A theoretical proof of the aforementioned statement will be impossible to obtain, since the error probabilities are dependent on the original distribution, the clusters,

[1]It should be noted that since the cluster models are built only over their corresponding clusters, the distribution of instances that they represent is not the original distribution, but one represented by the corresponding clusters.

the various thresholds and the learning model, all of which are highly variable in practice. In the subsequent sections, we will conduct a series of experiments in order to validate our framework; later chapters will apply all three frameworks over real-world problems.

4.2 Experimental Framework

The data employed in all the experiments conducted in this chapter is the same as in Chapter 3, and thus, we refrain from repeating their description in this chapter. For details, we direct the reader to Section 3.2.

As in the previous chapter, the one-class classifiers employed are the autoassociator (AA) and the one-class support vector machine (ocSVM), and the performance measure employed is the geometric mean of the per-class accuracies (g-mean). We use k-means clustering and partitioning around medoids (PAM) as the clustering algorithms; the results are reported for the best performing clustering approach.

We introduce three different frameworks for improving one-class classification in this chapter, each dependent on the extent of available domain knowledge for discovering and identifying sub-concepts during training and testing. Consequently, we conduct multiple experiments on each dataset, applying all appropriate frameworks as applicable; this is elaborated in the following table. **C** refers to using complete knowledge of concepts, **F** refers to the fuzzy knowledge representation and using multi-class classifiers to identify concepts, and **N** refers to the clustering approach for sub-concept discovery due to no knowledge of concepts.

Given that we created the artificial datasets, we are able to test all three frameworks over them; we can only test the framework for employing full knowledge over the artificial datasets as we can always determine which concept novel instances belong *a priori*, *i.e.* during both training and testing. Concepts are represented by the target and outlier modes that are closest to each other. In the diabetes, heart disease, ionosphere, thyroid disease and sonar datasets, we have no knowledge of any concepts, and thus, we can only implement the clustering framework over them. The

alphabets and forest cover derivative datasets were manually converted into binary problem for one-class classification by combining classes, as described in the previous chapter. Therefore, we treat each class utilized to compose the target class as a unique concept, and test the fuzzy knowledge framework over them. Note that we cannot test the full knowledge framework as we have no way of determining *a priori* which concept novel instances will belong to during testing. However, since we have that knowledge during training, we can test the fuzzy knowledge framework. While it may seem odd that, for example, an **A** is being classified as either of **M**, **N**, **I** or **J**, the idea is that we view all alphabets as being influenced by an underlying concept that is characteristic of the four target sub-concepts. Thus, when we observe an A, the classifier simply asks: relative to the four target sub-concepts, which one is A most influenced by. While this formulation may not represent precisely what would be encountered in practice, it does serve to illustrate the core concepts discussed in this chapter.

It is prudent to stress that all these UCI datasets are inherently binary classification problem; we include them for testing our framework for the sake of completeness. Furthermore, with the exception of the thyroid disease, alphabet and forest cover derivative datasets, none of these domains exhibit characteristics that would be typical of a one-class classification problem: firstly, there is enough balance in the data to warrant the application of a binary classification problem. Secondly, in a one-class classification problem, there is an abundance of data that can be employed to induce a one-class classifier. As we can see in Table 3.1, that is not the case for these domains; there are only between a 100-500 target samples to train from. Given these characteristics then, these few domains represent perhaps the worst case scenario that could be encountered in a one-class classification setting.

The number of clusters for the multimodal artificial dataset was set to 4, given the nature of the dataset. For the unimodal artificial dataset, the number of clusters ranged from 2 to 10. For all other datasets, they varied from 2 to 20. In the following section, we present the results obtained from conducting the experiments described in this section.

Table 4.1: Frameworks employed for each dataset

Dataset	**C**	**F**	**N**
Artificial Multimodal (NO)	✓	✓	✓
Artificial Multimodal (O)	✓	✓	✓
Diabetes	×	×	✓
Heart Disease	×	×	✓
Ionosphere	×	×	✓
Thyroid Disease	×	×	✓
Sonar	×	×	✓
Alphabets	×	✓	✓
Forest	×	✓	✓
ForestC1	×	×	✓
ForestC2C5	×	×	✓

4.3 Results

This section presents the results over the various datasets, displayed as barplots. The plots display results for both the autoassociator (AA) and one-class support vector machine (ocSVM). Depending on the dataset and the frameworks applied over it, each bar represents the g-mean value for the following:

Regular : The g-mean value for a regular one-class classifier (AA or ocSVM), with no framework applied.

Clusters : The g-mean value for a one-class classifier (AA or ocSVM) trained under the *no knowledge* framework (*i.e.*, with clustering).

Fuzzy : The g-mean value for a one-class classifier (AA or ocSVM) trained under the *fuzzy knowledge* framework.

Complete : The g-mean value for a one-class classifier (AA or ocSVM) trained under the *complete knowledge* framework.

MCC : The g-mean value of the *best* performing binary classifier, under the most extreme imbalance (*i.e.,* the highest imbalace ratio we used in the previous chapter), complimented with sampling. These are the same results from the previous chapter, and we present them here so that, along with the regular

one-class classifiers, we can compare the performance of the various knowledge frameworks and if they offer benefits over the state-of-the-art.

We begin with the artificial multi-modal datasets, followed by the UCI datasets.

4.3.1 Results on *Artificial Multimodal* datasets

The performance values for different frameworks for the multimodal datasets displayed in Figure 4.7 for the dataset with no overlap and in Figure 4.8 for the dataset with overlap.

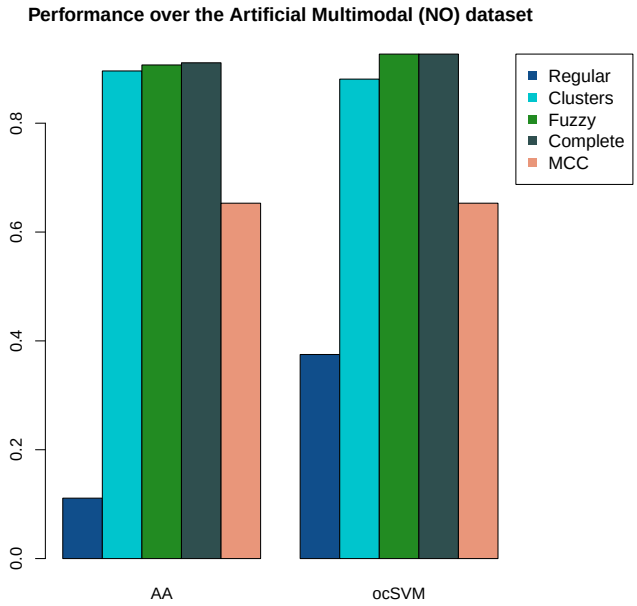


Figure 4.7: The performance values of various framework over the artificial multimodal dataset with no overlap.

The distribution is very complex, and thus it is no surprise to see that when the one-class classifiers are learning on the entire domain, their performance is very poor. However, when they are trained under our proposed frameworks, in all frameworks, their performance improves tremendously. Given that the modes are mostly distinct, clustering yields good results, as accurate clusters are identified. Furthermore, the Naïve Bayes classifier gives perfect classification, and thus the fuzzy approach performs very well. Finally, if novel instances are only classified using the target/outlier

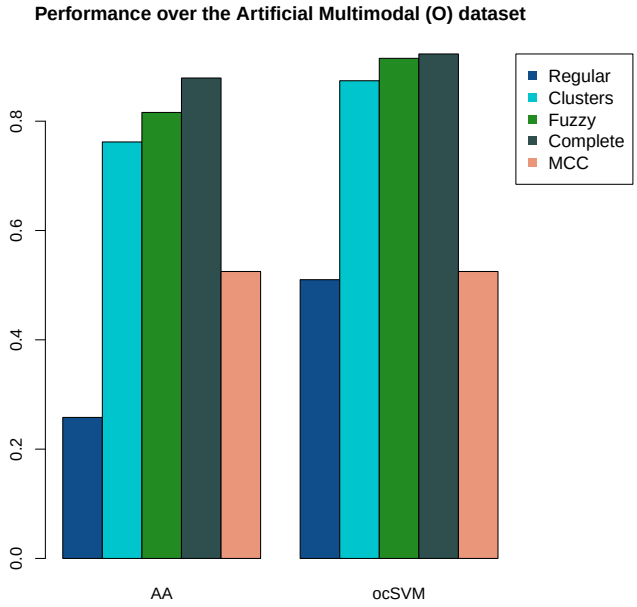


Figure 4.8: The performance values of various framework over the artificial multimodal dataset with overlap.

mode combination (*i.e.* the concept) to which they belong, it is no surprise that the performances are the best. This relative improvement is more evident in the case of the distribution with overlap; overlap tends to reduce the performance of classifiers and clustering algorithms, and thus, if we have access to complete knowledge of the concept to which an instance belongs to, we would expect that the performance of the resulting one-class classifier would be the best.

4.3.2 Results on *Diabetes* dataset

Performance of the classifiers for the Diabetes dataset are displayed in Figure 4.9.

This is a complex domain due to a significant overlap between the classes, and as outlined in the previous chapter, even with sufficient balance, binary classifiers do not perform very well. This complexity, and the general lack of data to begin with explains why one-class classifiers do not perform well. However, with clustering we notice a sharp increase in performance for both one-class classifiers. The performance improvement is most prominent for the autoassociator; it was initially worse than the

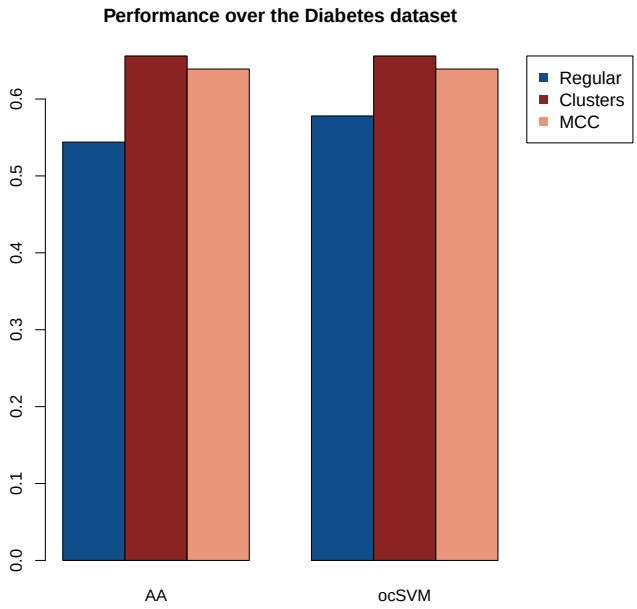


Figure 4.9: The performance values of various classifiers over the diabetes dataset.

best binary classifier but, with clustering, it outperforms it.

4.3.3 Results on *Heart Disease* dataset

Figure 4.10 displays the values of the classifiers over the heart disease domain.

This dataset is highly multi-modal, with small compact clusters. Both one-class classifiers perform better than the best case binary classifier, even without clustering. Given the nature of the dataset, however, we note an increase in performance. Note that while there is indeed an increase in performance, it is not as dramatic as one would expect, given the multi-modal nature of the dataset. This is due to the fact that there are only 75 instances in total from the target class available to induce a one-class classifier. Thus, as more and more clusters are formed, the amount of data necessary to induce a strong one-class decreases. Thus, we observe that while the performance does increase, the increase would perhaps be more pronounced if the amount of data available was greater.

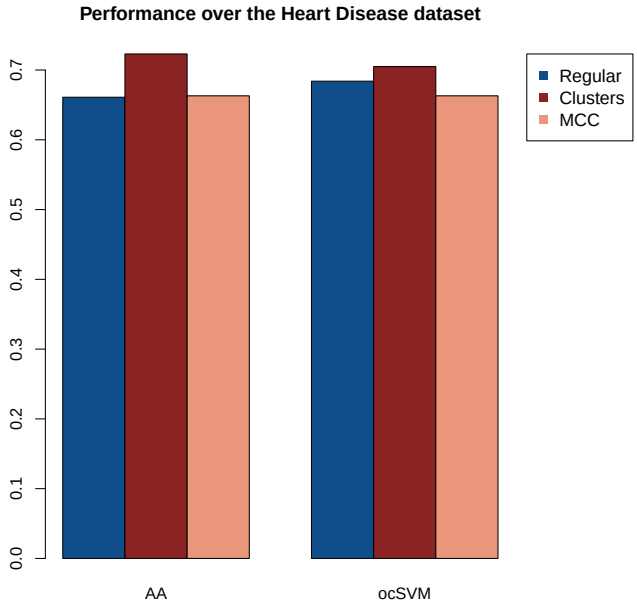


Figure 4.10: The performance values of various classifiers over the heart disease dataset.

4.3.4 Results on *Ionosphere* dataset

Performance values for classifiers for the Ionosphere dataset are displayed in Figure 4.11.

There is not a high level of overlap between the classes in this domain. Thus, both the AA and ocSVM have high performance values even without clustering. Clustering the domain, however, still leads to an improvement in performance, with the AA exhibiting a relatively larger gain. Both classifiers are significantly better than the best binary classifier; this is expected given the relatively simplistic nature of the domain.

4.3.5 Results on *Thyroid Disease* dataset

The values for classifier performance the for thyroid disease dataset are shown in Figure 4.12.

The outlier data in this domain is "overshadowed" by the target data (as discussed

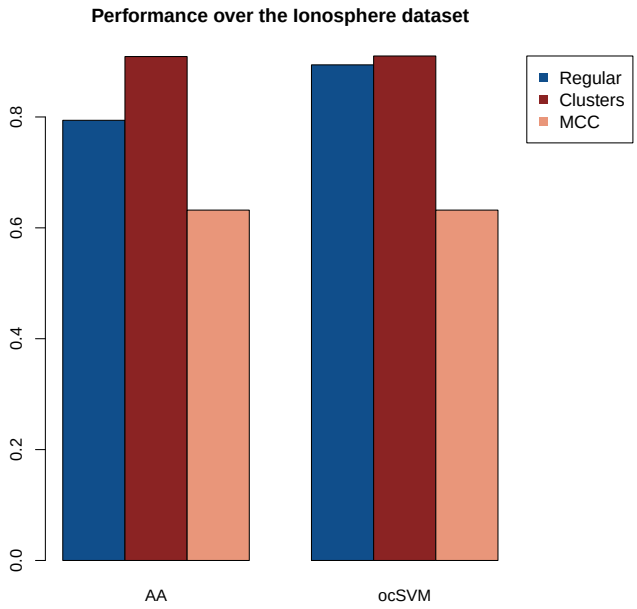


Figure 4.11: The performance values of various classifiers over the ionosphere dataset.

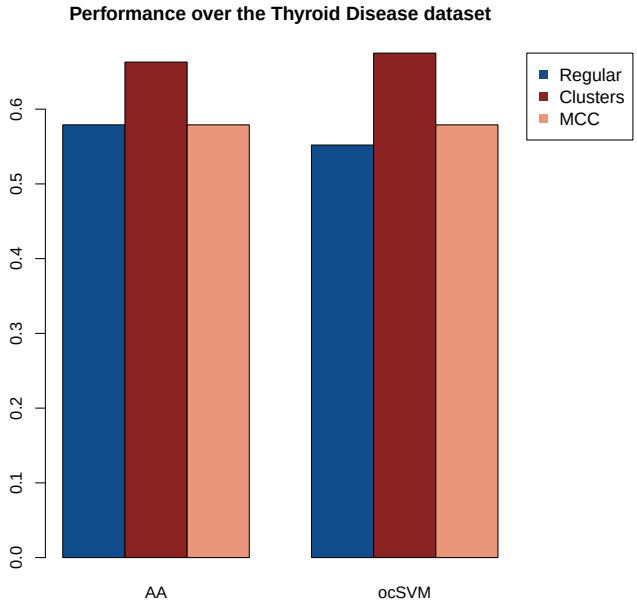


Figure 4.12: The performance values of various classifiers over the thyroid disease dataset.

in the previous chapter). Thus, the one-class classifiers do not perform very well; indeed they are worse than the best binary classifier. However, when clustering is applied, their performance improves significantly, surpassing their binary counterpart by a wide margin.

4.3.6 Results on *Sonar* dataset

Classifier values for the sonar dataset are shown in Figure 4.13.

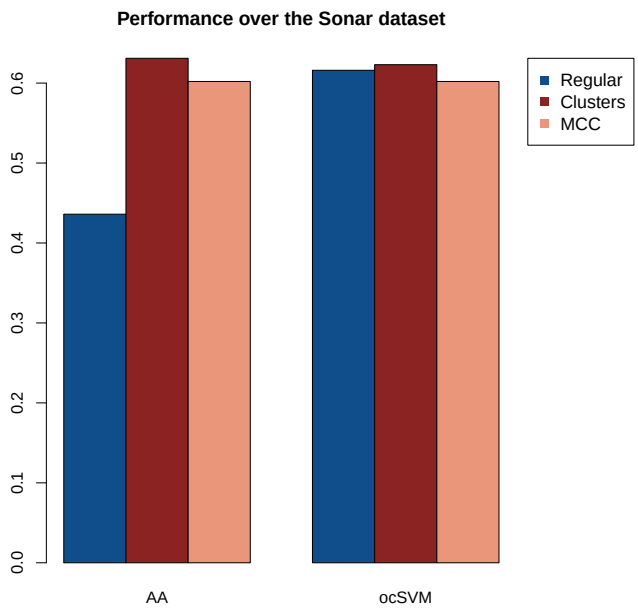


Figure 4.13: The performance values of various classifiers over the sonar dataset.

This dataset is perhaps the worst candidate for one-class classifiers due to high dimensionality and the general lack of data from both classes; there are only 55 target instances to learn from, whereas there are 60 dimensions. The ocSVM still manages to output a decent performance, though the AA suffers tremendously; it is significantly worse than the best binary classifier. Clustering does manage to alleviate performance for both classifiers over the binary classifer, with the improvement being most significant for the AA.

4.3.7 Results on *Alphabets* dataset

Figure 4.14 display the values for various frameworks over the alphabets dataset.

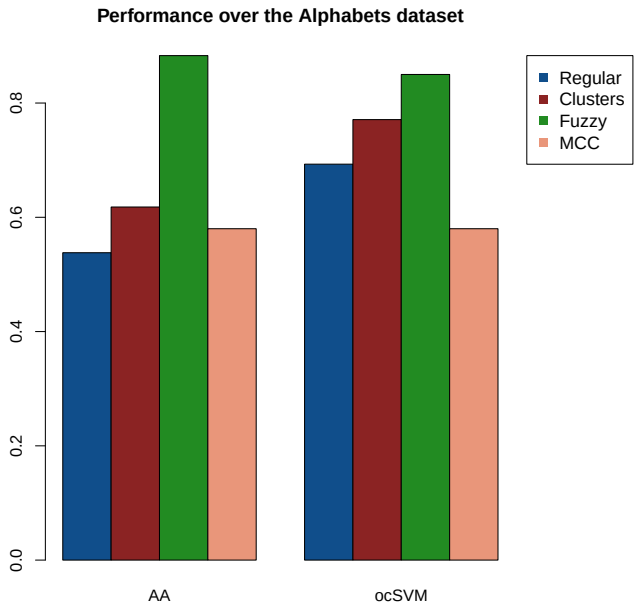


Figure 4.14: The performance values of various classifiers over the alphabets dataset.

There is a significant overlap between the target and outlier class, and there are three modes, one which has the bulk of the data. The overlap and multimodality causes the one-class classifiers to exhibit relatively poor performance; indeed, the AA is outperformed by the best binary classifier. Given the multimodality, it is unsurprising that we see an increase in performance when we cluster. However, the performance increase is significantly greater when we train the classifiers under the fuzzy knowledge framework. Specifically, a single one-class classifier is trained for each of the target class concepts, namely **M**, **N**, **I** and **J**. Furthermore, a multi-class classifier is trained to discriminate between each of these concepts. These concepts are highly discriminable; we employ the 1-nearest neighbour classifier, and it achieves a g-mean of 0.975. Novel instances are then classified by this classifier and passed for further processing by the classifier corresponding to the classified concept. As we are dealing with well defined sub-concepts, and learning and classifying explicitly over

them, we observe that the fuzzy knowledge framework yields the best performance.

4.3.8 Results on *Forest* dataset

The values for the classifiers on the forest dataset are displayed in Figure 4.15.

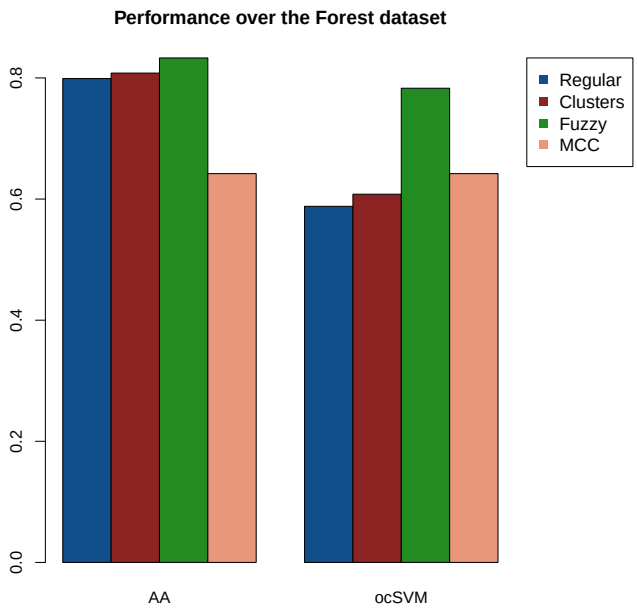


Figure 4.15: The performance values of various classifiers over the forest dataset.

The target space is composed of four sub-concepts, three of which are very similar to each other, and exist as a single mode. The fourth concept is distinct, and thus, the overall space exists as a bimodal distribution. With respect to the outlier space, the target class is almost completely overwhelmed with the outlier class in most dimensions, though it does discriminate itself from the outlier class in some dimensions. Given the inherent complexity of the target space, it is easy to see that both the fuzzy knowledge and clustering approaches yield superior performances over the binary classifier, as well as the one-class classifiers trained on the whole space. Furthermore, since the space is bimodal, and the classifier trained for the fuzzy knowledge framework performs with high accuracy (we employ a decision tree that yields g-mean of 0.881), the fuzzy framework outperforms the clustering framework.

4.3.9 Results on *ForestC1* dataset

The values for classifiersover the forestC1 dataset are shown in Figure 4.16.

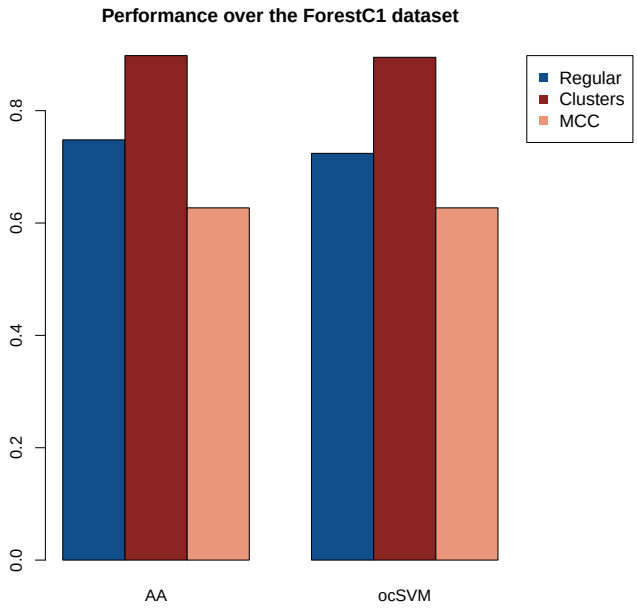


Figure 4.16: The performance values of various classifiers over the forestC1 dataset.

This domain, while being rather simple, exhibits significant overlap between the two classes. The simplicity of the domain is evident in the base performance of both one-class classifiers, as both the AA and ocSVM perform relatively well. Given the high level of overlap, however, we observe that clustering is able to significantly improve the performance of both one-class classifiers; the resulting classifiers significantly outperform the best one-class classifier.

4.3.10 Results on *ForestC2C5* dataset

Figure 4.17 shows the classifier values over the forestC2C5.

Despite there being two different tree species within the target class, the high level of similarity between the two results in a unimodal distribution. The extent of overlap between the target and outlier classes is, however, less severe compared to the forestC1 dataset. As a result of the simplicity, the base performance of both classifiers

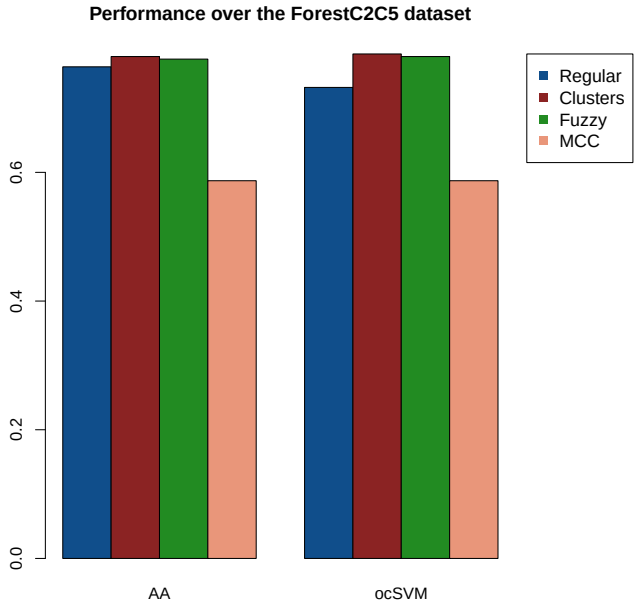


Figure 4.17: The performance values of various classifiers over the forestC2C5 dataset.

is relatively high, and given that the level of overlap is less, both classifiers perform slightly better when compared to the classifiers over the forestC1 domain. Once again, due to the high level of overlap, both clustering the domain and employing fuzzy knowledge yields performance improvements. These improvements are not as significant when compared to the other forest cover derivatives; this is due to the characteristics of the domain. Specifically, the domain is unimodal (compared to forest, which is multimodal), and the overlap, while large, is not as severe (compared to forestC1).

4.3.11 Statistical Analysis

For completeness, we perform a statistical test on the results obtained using a one-sided Wilcoxon Signed Rank Test. A significance level of 0.05 was selected. We begin with tests performed over the regular and clustered versions of the one-class classifiers, presented in Table 4.2. Table 4.3 presents the results between the regular and fuzzy knowledge versions. In both cases, we note that there is a statistically significant

Table 4.2: Results of the Wilcoxon Signed Ranks test for the clustered and regular versions of the one-class classifiers.

Classifier	p-value	α-value	Significant?
Autoassociator	0.0004	0.05	**yes**
One-class SVM	0.001	0.05	**yes**

Table 4.3: Results of the Wilcoxon Signed Ranks test for the fuzzy knowledge and regular versions of the one-class classifiers.

Classifier	p-value	α-value	Significant?
Autoassociator	0.03	0.05	**yes**
One-class SVM	0.03	0.05	**yes**

difference in performance between one-class classifiers trained over the whole domain, and those that employ some form of domain specific knowledge to boost performance.

4.4 Discussion

The experiments conducted were intended to investigate and analyze how the frameworks introduced in this chapter could be applied across a broad spectrum of domains, with varying levels of complexity. In all cases, we observe that there are improvements in classifier performance when they trained and applied within any one of the proposed frameworks; the extent to which we observe an improvement is, naturally, dependent upon the complexity of the domain.

Table 4.4 summarizes the results over all domains; we list the g-mean for the regular version of each one-class classifier, along with the g-mean of the one-class classifier under the best performing framework (**AA** refers to the autoassociator, **K-AA** refers to the best performing knowledge framework for the autoassociator, **ocSVM** refers to the one-class SVM and **K-ocSVM** refers to the best performing knowledge framework for the one-class SVM). Furthermore, we present the g-mean of the binary classifier under the most extreme imbalance, complimented by the best performing sampling method from the previous chapter (denoted by **MCC**); this serves to illustrate how our framework compares to the state-of-the-art in binary

Table 4.4: Summary of results over all domains

Dataset	**AA**[1]	**K-AA**[2]	**ocSVM**[3]	**K-ocSVM**[4]	**MCC**[5]
Artificial Multimodal (NO)	0.111	0.911	0.375	0.927	0.653
Artificial Multimodal (O)	0.258	0.879	0.510	0.923	0.525
Diabetes	0.544	0.656	0.578	0.656	0.639
Heart Disease	0.661	0.723	0.684	0.705	0.663
Ionosphere	0.794	0.909	0.894	0.910	0.632
Thyroid Disease	0.579	0.663	0.552	0.675	0.579
Sonar	0.436	0.631	0.616	0.623	0.602
Alphabets	0.538	0.883	0.693	0.850	0.580
Forest	0.799	0.833	0.588	0.783	0.642
ForestC1	0.748	0.898	0.724	0.895	0.627
ForestC2C5	0.764	0.780	0.732	0.784	0.587

[1] Autoassociator
[2] Best performing knowledge framework with the autoassociator
[3] One-class SVM
[4] Best performing knowledge framework with the one-class SVM
[5] Best performing binary classifier with sampling

classifiers and sampling. In all domains, we note that employing knowledge improves the classification, and that the best performing binary classifier is outperformed by the one-class classifiers under the appropriate framework. This is especially important in cases where the binary classifier, trained with sampling under the most extreme imbalance, still manages to perform at par, if not better, than a one-class classifier. The domains in which this occurs for both one-class classifiers are the two multi-modal artificial domains and the Diabetes and Thyroid Disease domain, whereas in the Alphabets, Sonar and Heart Disease domains the binary classifier outperforms only one of the one-class classifiers. In all cases, the domains exhibit complexity in terms of high overlap (all domains) and multi-modality (all except diabetes and sonar). As we have highlighted previously, the advantage one-class classifiers offer over binary classifiers disappears if the domain is complex, especially in terms of overlap and multi-modality; the employment of domain knowledge, however, can mitigate this. This is precisely what we observe in these domains.

The fuzzy knowledge framework is highly dependent upon the power of the classifier that is utilized to discriminate between the various sub-concepts; the more

powerful the classifier, the greater the improvement that we will see. This is evident in the improvements observed over the alphabets domain and the forest and forestC2C5 domains. The classifier employed in the alphabets domain has a g-mean of 0.975, and thus, we observe a significant improvement. The classifiers for the forest and forestC2C5 datasets have g-means of 0.880 and 0.915, respectively, and thus, we note less of an improvement over the baseline.

Before we conclude, it is worth considering where exactly the improvement in performance by employing knowledge is coming from. We revisit the discussion from Section 4.1, where we noted that learning without taking implicit knowledge into account can lead to over-generalization. The implication of this is that the majority of outlier data would be misclassified as belonging to the target class, and thus the accuracy over the outlier class would be very poor. Employing knowledge can mitigate the detrimental effect of over-generalization. In the experiments discussed in this chapter, we observe that the improvement in performance comes entirely from an increase in the power of the one-class classifier to correctly classify novel instances as outliers (*i.e.*, an improved outlier detection accuracy). In Table 4.5, we display the per-class accuracies for both the target and outlier classes, for the case when the one-class classifier employs no knowledge (under the column **NK**), and for the best performing knowledge framework (under the column **K**).

All domains exhibit an increase in the outlier class accuracy; in some cases, the increase is highly significant. In some domains, the true positive rate declines slightly; this is due to the classifier not over-generalizing over the complex domain. A simple classification rule would be to accept everything as belonging to the target class, and given that we only have data from that class, this would ensure perfect accuracy over the available data. In other words, we over-generalize over the available data. The more complex the domain, the higher the likelihood of such an over-generalization happening. While this would ensure high accuracy over the target data, the performance over novel outlier instances would be dismal. This is evident from the results presented; all domains exhibit some level of complexity, and we observe that the outlier accuracies are relatively low compared to the target accuracies. Employ-

Table 4.5: The target and outlier class accuracies over the various domains

Dataset	Classifier	**Target Accuracy**		**Outlier Accuracy**	
		NK	K	NK	K
Artificial Multimodal (NO)	AA	0.946	0.961	0.029	0.849
	ocSVM	0.894	0.861	0.157	0.999
Artificial Multimodal (O)	AA	0.950	0.941	0.074	0.827
	ocSVM	0..893	0.857	0.291	0.995
Diabetes	AA	0.924	0.703	0.308	0.656
	ocSVM	0.809	0.640	0.421	0.642
Heart Disease	AA	0.898	0.748	0.490	0.628
	ocSVM	0.730	0.654	0.643	0.737
Ionosphere	AA	0.950	0.926	0.680	0.890
	ocSVM	0.930	0.912	0.860	0.890
Thyroid Disease	AA	0.658	0.737	0.509	0.596
	ocSVM	0.697	0.593	0.444	0.768
Sonar	AA	0.690	0.590	0.270	0.642
	ocSVM	0.645	0.488	0.584	0.791
Alphabets	AA	0.947	0.925	0.307	0.843
	ocSVM	0.886	0.840	0.543	0.860
Forest	AA	0.897	0.886	0.712	0.787
	ocSVM	0.868	0.896	0.385	0.685
ForestC1	AA	0.893	0.876	0.626	0.920
	ocSVM	0.897	0.892	0.584	0.897
ForestC2C5	AA	0.890	0.844	0.649	0.720
	ocSVM	0.898	0.908	0.597	0.677

ing domain knowledge to identify and learn along sub-concepts prevents this sort of over-generalization from happening. While the target accuracy may go down slightly, by tightening the classifier, we are correctly able to reject a much larger number of outliers, which, especially in security-based domains, is extremely crucial!

4.5 Summary

Thus far, we have considered the contexts under which one-class classification becomes the sole solution for learning purposes. For it to be effective, a one-class classifier must be able to model the data from the target class as precisely as possible, a task that can be difficult when dealing with complex distributions. In order to alleviate this issue, we investigated the concept of intelligent sub-division of complex distributions into simpler distributions, where the simpler distributions correspond to sub-concepts within the domain. Depending on the depth and extent of knowledge available, we introduce three different frameworks. We illustrate the power of these frameworks by testing them on artificial datasets and datasets from the UCI Repository. The results show that there is, indeed, an improvement in performance of one-class classifiers, and this is reaffirmed by statistical analysis done using the Wilcoxon Signed Ranks Test.

Simply testing on artificial and UCI datasets, however, is not enough. These frameworks must work in real-world, practical domains. In the following chapters, we demonstrate the application of each of these frameworks in domains that we encountered during the course of our research. Each of these domains conform to the one-class classification framework, are complex, noisy, and ideally suited for validating the frameworks proposed in this chapter.

Chapter 5

Biometrics for Security

In the previous chapters, we analyzed the general conditions under which a domain becomes suitable for one-class classification and provided a framework for improving the performance of one-class classifiers when dealing with domains with complex distributions. In this and subsequent chapters, we will apply these insights to real world problems. In this chapter, with look into the domain of biometric authentication on mobile phones. Specifically, the task is to employ a users *swipe* across the touch screen of a mobile device to provide authentication; the choice of employing swipes as a means of authentication is a novel technique resulting from our work, deviating from standard methods such as PINs, passwords, patterns, and fingerprints. The different sub-concepts arising in this domain are based on a user's motion, as it was observed that the data recorded by the sensors on the phone when users were mostly stationary different significantly from when they were in motion. Consequently, the level of security that was offered by swipes improved vastly when we conducted training and classification on these implicit sub-concepts. Modern phones are capable of detecting whether a user is in motion, and thus, the work presented in this chapter falls under the first framework introduced in the previous chapter, namely, employing complete knowledge of sub-concepts for improving one-class classification.

We begin with an overview of the field of biometrics in Section 5.1, followed the various behavioural biometrics currently being researched for mobile phones in Section 5.1.1. Section 5.2 introduces the notion of employing swipes as a behavioural

biometric for authentication. In particular, we highlight why a users swipe represents a convenient solution, along with the implementation details regarding the swipe representation. The learning challenges associated with developing a practical solution are then introduced in Section 5.3, followed by the learning algorithm in Section 5.4. The experiments and associated results are discussed in Section 5.5 and 5.6 respectively. We conclude the chapter in Section 5.7.

5.1 Biometrics: An Overview

The need for reliable automated identity verification has become an important facet of our daily lives; it comes into play within a wide variety of situations, from logging into devices, checking emails, to conducting sensitive financial transactions. The task of verification can be viewed as being composed of two separate actions: the first being an authorization mechanism that verifies the identity of the user, and the second a mechanism that decides upon appropriate actions to take depending on the result of the first action. With respect to the former, Woodward *et. al* outline three different types of mechanisms [85], each relying on a unique aspect of the user:

- Something one *knows*: This refers to some *secret knowledge* that a user possesses, such as passwords or PINs.
- Something one *has*: This refers to some **physical object** that a user possesses, such as a SIM card or a token.
- Something one *is*: Also referred to as *biometrics*, this refers to an intrinsic physical characteristic, or a personal trait, that a user possesses, such as a fingerprint or gait.

Biometrics offer two key advantages. First, they are convenient as there is no danger of forgetting (eg. a password) or losing (eg. an access card) the authentication key. Secondly, passwords and cards can be easily stolen, whereas given biological diversity, it is highly unlikely that two people will share the same biometric signatures.

The utilization of biometrics (literally translated as "*life measurement*") for authentication and identification purposes is not solely a product of the digital age. Over a thousand years ago, fingerprints were imprinted upon wares by East Asian potters for brand identity, whereas physical features such as height and complexion were used for formally identifying Nile Valley traders [85]. However, it was not until the nineteenth century that the first biometric technique was formally described, motivated by the need for identifying and capturing recidivists. In France, Alphonse Bertillon developed *anthropometrics*, a method of taking multiple physical measurements and unique and peculiar characteristics, such as scars, birth marks, tattoos etc., of a human being for the purposes of identification [85].

Biometric authentication has come a long way since it's humble beginnings; advances in technology has made it possible to employ a vast array of biometrics such as voice, eyes, facial features, gait and typing characteristics. Given the number of different biometrics that are available, it is prudent to categorize them based on which biological factor they employ. Woodward *et. al* [85] identify three distinct areas:

Genetic Biometrics : These are inherited features that are a result of an individuals genetic makeup. Examples include eye colour, hair colour and facial structure.

Phenotypic Biometrics : These are features created during early stages of embryonic development. Iris patterns, fingerprints and vascular networks are examples of these features.

Behavioural Biometrics : These features capture the subtleties and nuances of an individuals behaviour with respect to certain activities. Behavioural biometrics are typically learnt over time. While some have the propensity to change, such changes tend to be temporary. Handwriting, typing, voice and gait are examples of behavioural biometrics.

The system we develop in this chapter for securing mobile phones relies on *behavioural biometrics*. Yampolskiy *et. al* [91] categorize behavioural biometrics into

five distinct groups. The first group consists of metrics that are based upon authorship; users are identified based on the uniqueness of text, drawings, vocabulary etc. Human computer interaction (HCI) based biometrics [88] make up the second group. Authentication is performed by monitoring how users interact with input devices such as keyboards and computer mice. The third category comprises of indirect HCI-based biometrics [89], which include traces of system call, program execution trees etc. While the second and third group may appear similar, the difference lies in the fact that the interaction in the second group is more explicit as opposed to the third group, where it is more subtle and implicit.

The fourth, and perhaps the most researched group, consists of motor-skill based biometrics [90]. These include popular biometrics such as voice, signature and keystrokes. Finally, the fifth and final category consists of biometrics that are purely behavioural in nature. Specifically, these biometrics quantify subtle traits that human beings employ while solving mentally demanding tasks.

While biometrics have been employed for a number of tasks such as intrusion detection, verification of users on desktop machines, and fraud detection, their use for authentication of users on mobile devices is relatively new. In the following section, we review the current state-of-the-art in the employment of behavioural biometrics for security purposes on modern smartphones.

5.1.1 Behavioural Biometrics in Smartphones

The proliferation of touch screen based smartphones opened up the gates to a range of possible research opportunities, as researchers could harness not just keystrokes but patterns and gestures that users made on the screens. Initial work by Kolly *et. al* [45], De Luca *et. al* [57] and Frank *et. al* [22] aimed at looking at the possibility of using the touch screen features returned by the sensors for discriminating between the user and an impostor. Kolly *et. al* [45] examined a large database comprised of over 1 million touch button events of $14,890$ users. Experiments were conducted on the extracted features with the aim of gaining insight from two perspectives: inter-user discriminability (classification) and user authentication (anomaly detection). By

using the Naiv̈e Bayes classifier, they obtained, for the former, recognition rates of 94% between 2 users; it went down to 60% for 10 users. For the latter, they reported an Equal Error Rate (EER) of 30%.

The work by Luca *et. al*, [57] was based upon the Android screen lock patterns; their aim was to add behaioural biometrics from the touch screen as an extra layer over the screen lock patterns. With respect to modelling, they undertook a deviation from the norm by examining the raw time series returned by the touch sensors, rather than discretizing it into discrete, real valued features. They employed the Dynamic Time Warping distance (DTW) to compare the distances between novel and exemplar time series, and used that as a means of authentication. Their user study reported an False Alarm Rate (FAR) and False Rejection Rate (FRR) of 21% and 19% respectively.

Finally, the authors in [22] were interested in looking at touch screen gestures for continuous authentication. In particular, a collective authentication decision is made every 11 consecutive touch gestures. The enrollment phase captures and records data until a representative set is gathered, and a model is built using k-nearest neighbour and a radial basis support vector machine (SVM). Their study reported EER's between 0% and 4%, based on the usage scenario.

As was the case with keystrokes, once initial research showed promise of using touch screen gestures as a biometric for security, research into the field expanded, with different feature representations, algorithms and devices being explored. For instance, Meng *et. al.* [61] introduced the concept of sessions, a unique temporal set having features characteristic to it, such as the number of touch gestures per session. In their work, they defined a session as being 10 minutes long. On a user study with 20 users, they reported FAR and FRR of 7.8% and 7.08% respectively, which improves to 2.5% and 3.34% by optimising the network with particle swarm optimization (PSO). Saravanan *et. al.* [67] analyzed their framework on both mobile phones and tablets, reporting results from both a user recognition and user discrimination perspective (the later being tested for shared devices). Using a one-class SVM for the former, they reported an average accuracy of 97.9% on the phone and 96.79% for a tablet. When testing for user discrimination, over 5 users they obtained perfect accuracy for

a tablet and 97.78% accuracy on a phone.

As the technology in smartphones improved, researchers noted that the notion of *implicit authentication*, as introduced by [36], would be best achieved by combining the outputs of multiple sensors such as accelerometers, gyroscopes, pressure sensors, GPS etc. Our work considers the action of a users swipe across a touch screen as an implicit authentication mechanism. In the next section, we introduce the notion of using swipes as a means for authentication.

5.2 Swipes as a behavioural biometric for mobile phones

The first step in our work for developing an implicit authentication system was to select a biometric that satisfies three key features. First, it has to represent an action that is very *common*. Specifically, it must be an action that users perform so often that it becomes almost second-nature. This underlines the notion of *implicit*. Secondly, it must exhibit an ample level of *regularity*. This is important from a learning perspective; learning can only occur if the data displays a reasonable level of consistency. Finally, it must be *discriminative*. This is key from a security perspective, as users must possess a uniqueness for that biometric.

A behaviour that satisfies all these features is the horizontal sliding away of a screen. Users typically perform this task while switching between different screens (such as in a gallery, different application screens, etc.), different items (such as in an RSS reader), and even to unlock the phone (the iOS lockscreen). An example of this interface is presented in Figure 5.1 for two popular phones, the iPhone and the Samsung Galaxy Note. Given that the action typically occurs at a subconscious level, swipes for a specific task tend to display an adequate level of regularity. Furthermore, given the significant latitude that users have in swiping in their own distinct manner, swipes are highly discriminative.

When a user swipes across the screen, each sensor generates a time series; the

(i) (ii)

Figure 5.1: Lock screens that employ sub-conscious swipes for the Samsung Galaxy Note in (*i*) and the iPhone in (*ii*).

touch screen time series represents the Cartesian co-ordinates of the swipe across the screen at different time intervals, and the accelerometer and gyroscope time series represent the motion of the phone in 3-dimensional space while the swipe is being done across the screen. We hypothesize that these time series exhibit unique patterns and nuances across users, and by appropriately modelling them, it is possible to utilize swipes as an authentication mechanism. Indeed, during our visual analysis of raw data, this observation was validated. Consider the plot in Figure 5.2; it shows the plot of x-axis accelerometer values for three different users. It is obvious that the time series are highly distinct for the users; the range of values that compose the time series, as well as the temporal trends represented by them, are both discriminatory. The challenge, thus, is to come up with an appropriate representation, as well as a one-class classification framework tailored to the domain, in order to develop the authentication system. The representations are described in detail in the following sub-sections.

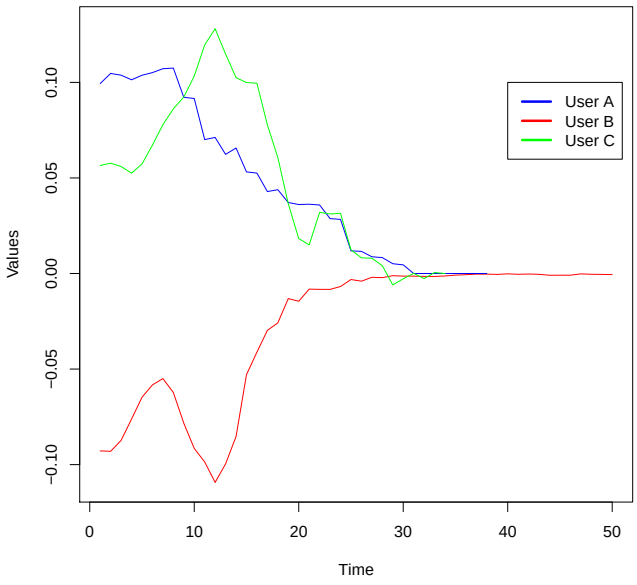


Figure 5.2: x-axis accelerometer data for three users.

5.2.1 Swipe Representation

We being with the discretization of the touch features, followed by the discretization of accelerometer and gyroscope sensors.

Touch Screen Features

Figure 5.2.1 illustrates the geometry of a swipe across a screen. The actual swipe consists of, as mentioned previously, a vector of cartesian coordinates that represent the physical curve. Using these coordinates, we are able to generate the following features:

Start and End coordinates : These are represented by the $(start_x, start_y)$ and (end_x, end_y) co-ordinates of the swipe, as shown in Figure 5.2.1.

Direction : This is the angle between a horizontal line across the screen and the line segment joining the start and end co-ordinates of the swipe. It is represented by θ in Figure 5.2.1.

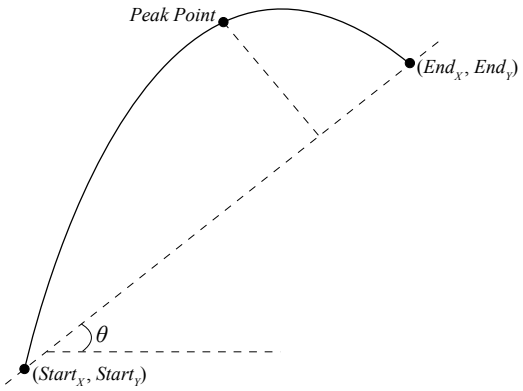


Figure 5.3: Touch Features

Magnitude : This is the length of the line segment, in pixels, between the start and end co-ordinates of the swipe.

Length : This is the length of the actual swipe curve.

Time : The duration, in milliseconds, of the swipe.

Peak Size : The peak represents the point on the swipe curve that has the greatest distance from the line segment between the start and end co-ordinates. This feature represents this distance, in pixels.

Peak Time : The time taken to reach the peak point from the start of the swipe.

Longest Time Interval : This is the longest length of time between two consecutive touch sensor readings in the corresponding time series.

Highest Velocity : The velocity is defined as the ratio of the distance of the curve segment in an interval of the time series to the duration of that interval. This value represents the ratio with the highest value over the time series.

Accelerometer and Gyroscope Features

The accelerometer and gyroscope also return time series, just as the touch sensor. However, the values returned by these sensors are not as easily comprehensible as they are for the touch sensor. Furthermore, they are in three dimensions, as the

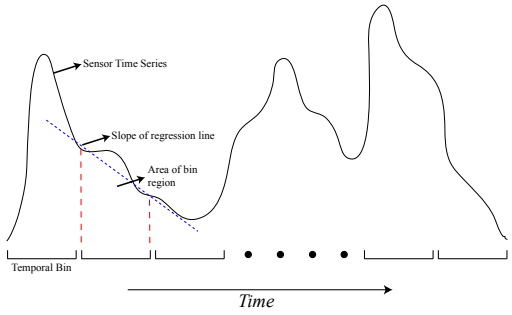


Figure 5.4: Discretization of the accelerometer and gyroscope time series.

sensors report values for the x, y and z axis. Given the obtuse nature of these values, we employ a more abstract methodology to convert their raw time series into usable features. What is of utmost importance while converting a time series into a set of discrete features is that the temporal information is preserved; ultimately, the kinetic motion of the device that we are trying to model is implicit in that temporal information.

Our discretization methodology is illustrated in Figure 5.2.1. We begin by binning the time series into a fixed number of bins. Thus, each bin will represent a curve that represents a fixed temporal section of the time series. For each bin, we calculate two distinct features: the *slope* of the regression line for the curve in the bin, and the *area* under the same curve. The slope represents the direction in which the time series travels in that particular time segment, whereas the area represents the magnitude of the motion. Together, these two values represent a fixed temporal section in the time series.

An application was initially developed in PhoneGap using JavaScript for the iPhone that allowed for swipe data to be recorded from the phone and sent to the data base. Eventually, the system was ported in C++ and became integrated into Android as well. The experiments and research done in this domain utilizes the Android application.

5.3 The Nuances of Swipes

In this section we will dissect the domain in terms of the two aspects alluded to in the previous chapters. We begin with a discussion regarding the imbalanced nature of the domain, followed by an analysis pertaining to the levels of complexity inherent within it. We conclude this section with general remarks on the challenges from a learning context that are specific to this domain. In particular, we discuss the challenge in obtaining a relatively consistent input from the user, as well as usability issues; in order to ensure that the application developed is user-friendly, certain challenges need to be addressed.

5.3.1 Learning Complexity

With a practical deployment of an application over this domain (for example, a user using swipes to act as a password replacement for a particular application), the only data available to learn from is the user's data. Data from *attackers* will be impossible to obtain and, if an attacker was genuinely using the application, it will be impossible to ascertain whether it was the attacker. This is because the user interface will not provide a means by which a swipe can be labelled to indicate if it belongs to the user or an attacker. Thus, during learning, we have to assume that all the swipes collected prior to induction are from the user.

Furthermore, given that each user has a unique biometric signature (a fundamental, key principle in the field of behavioural biometrics), even if we were to have access to a large pool of data from a plethora of users, a multi-class classifier induced over this space can only discriminate between the distributions represented by the user pool; it cannot generalize over the entire user space of all possible users.

Given these observations, then, the domain presents a pure one-class classification problem, conforming strongly to our domain characteristic of extreme imbalance as identified in Chapter 3. Next, we analyze the complex nature of the domain distribution.

5.3.2 Domain Complexity

Human behaviour is inherently noisy; data gathered as a result seldom comes from a neat, simple distribution. Consequently, swipes gathered from users through phones also exhibit a high level of variability. The most common source of this variability will typically be motion and posture. For example, a user may swipe the phone while sitting perfectly still at a desk; the behaviour captured by this swipe will be significantly different from a swipe done while walking at a brisk pace. In other words, the space of swipes, *i.e.*, the underlying distribution that generates all possible swipes by a user, will be comprised of multiple different sub-concepts (implicit concepts), each reflecting a distinct user motion. Note that we restrict our definition of these sub-concepts within the context of user motion; while it is likely that other contexts are possible, this definition is by far the simplest to visualize and identify through sensor outputs. Thus, in this thesis, we focus only on motion-specific behaviours.

Figures 5.5 and 5.6 illustrate the uniqueness of motion-specific data in this domain on two distinct users. We conduct principle component analysis on the transformed data (represented as previous described) , split by sensors, and plot the first three principal components. The *sitting* distribution forms a small, tight sub-concept within the larger, more spread *walking* distribution Intuitively, this makes sense, as sensors that are sensitive to motion will naturally produce a more spread distribution for motion. To further highlight the distinctiveness of data between different motions, Figures 5.7 and 5.8 and Figures 5.9 and 5.10 display the first three principal components for the both users individually.

In order to further analyze the distinctiveness of these sub-concepts, we perform an experiment over data of two users considered in our study using a binary classifier. In particular, we treat each motion as a unique class, and induced a Naïve Bayes classifier, with the aim to see how discriminatory each class is. The geometric means of the per-class accuracies (g-mean) are detailed in Table 5.1. We also include the result for training and testing the classifier over a combined dataset; this serves the illustrate that the sub-concepts exist across users. These results speak for themselves;

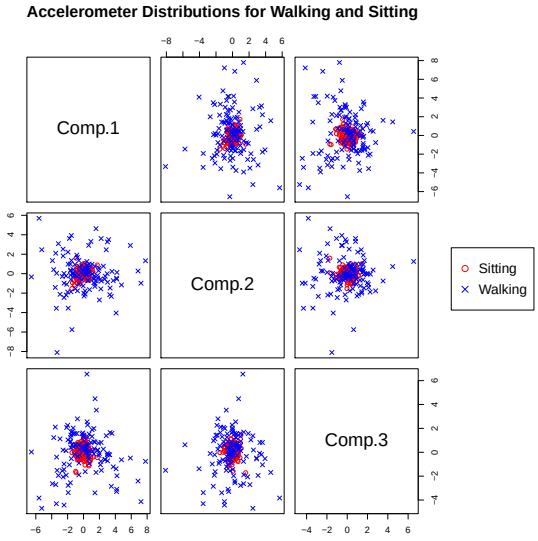


Figure 5.5: The first three principal components for the accelerometer features for Users 1 and 2.

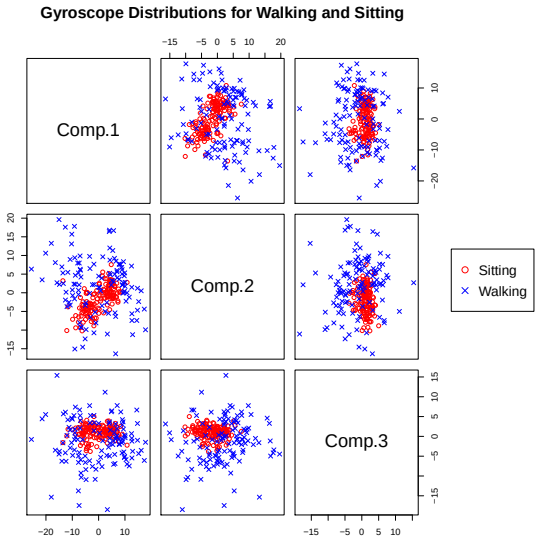


Figure 5.6: The first three principal components for the gyroscope features for Users 1 and 2.

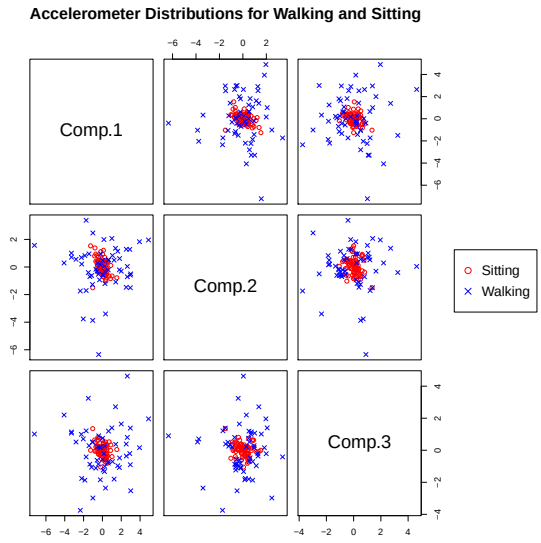


Figure 5.7: The first three principal components for the accelerometer features for User 1.

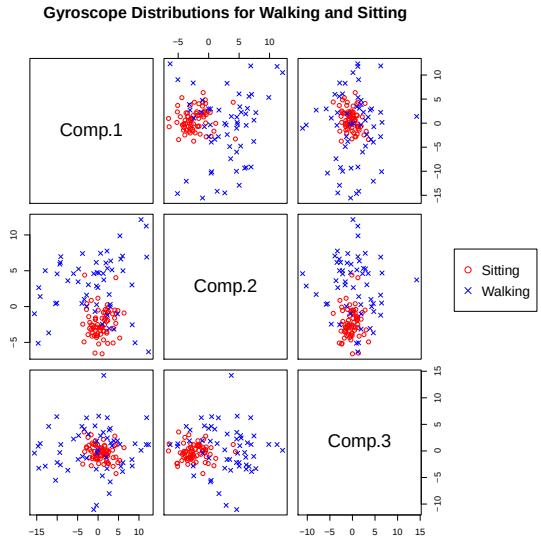


Figure 5.8: The first three principal components for the gyroscope features for User 1.

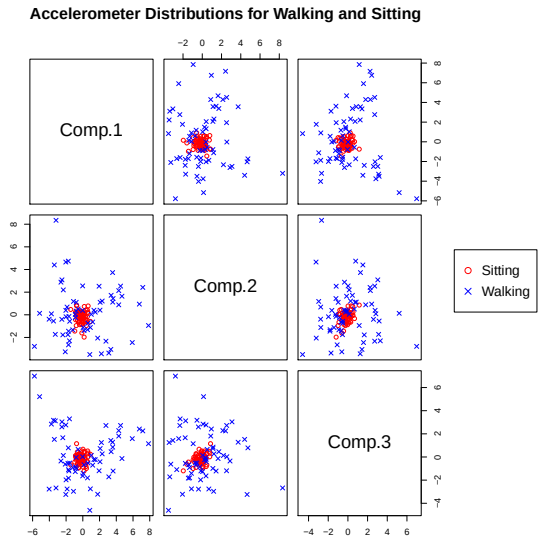


Figure 5.9: The first three principal components for the accelerometer features for User 2.

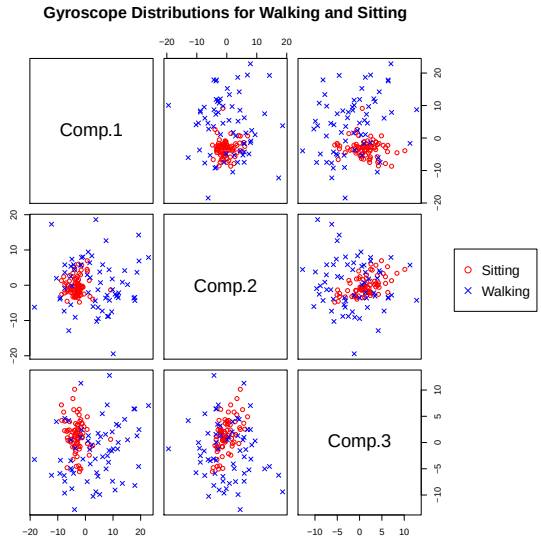


Figure 5.10: The first three principal components for the gyroscope features for User 2.

if the underlying distributions of the two motions were similar, then we would not obtain these values (the values would be closer to 0.5), as both classifiers would perform similarly.

Table 5.1: g-mean values for different users

User	**g-mean**
User 1	0.971
User 2	0.96
User 1 and 2	0.965

These observations, along with the visual analysis, highlight the pitfall in learning a single one-class classifier over this domain; the model will naturally over-generalize and lead to poor performance with respect to false positives (unauthorized users being authenticated).

5.3.3 Other Considerations

From a learning perspective, a practical deployment of an application over this domain presents a unique learning challenge. Firstly, a classifier needs to be learnt using very little data. This is due to the fact that it is impractical to ask the user to provide a large number of swipes prior to authentication; that would be highly inconvenient. Secondly, user behaviour for swipes tends to vary over time. These variations may either be permanent, resulting in a concept shift, or temporary, resulting in concept oscillation. These considerations necessitated research into building a novel learning and authentication algorithm, that will be detailed in subsequent sections.

A few other domain-specific points should be made for completeness. Naturally, a swipe generated by the user will be highly task-specific. For instance, the motion of browsing a website will be distinct from the motion employed for browsing through images in a gallery. Specifically, a user will not simply swipe without a purpose; it is the purpose that defines the nature of the swipe. Given this crucial observation, it is straightforward to ascertain that attempting to characterize the nature of swipes independent of their underlying purpose would be an extremely challenging task. As a result, we restrict the purpose to be analogous to an iPhone's *slide to unlock*. The

user simply swipes across the screen with the purpose of unlocking their phone. Given that the purpose is to make the authentication process implicit, we do not provide any visual cues or icons that need to be explicitly *swiped* away; the user simply swipes having assumed the purpose to be analogous to what they are familiar with. The experiments reported in this thesis rely on the swipe data gathered through this medium.

Summary

This section presented the characteristics of the swipe domain that have major implications on learning. We noted that the domain displays high levels of complexity that are a direct result of the users motion during a swipe. Furthermore, as it is impossible to know the source of the swipe *a priori*, we can only assume that all swipes obtained during the learning period are from the user. This implies that learning can only occur within a one-class classification paradigm. Thus, this domain is ideally suited for the framework outlined in this thesis. In remainder of the chapter, we will detail the learning framework employed, and the results and insights garnered.

5.4 Learning Framework

As described previously, a swipe, in its raw form, is comprised of 8 time series: the $x-$, $y-$ and $z-axis$ accelerometer time series, the x, y and $z-axis$ gyroscope time series and the x and $y-axis$ touch screen time series. These are discretized into temporal bins, with each bin being represented by the area of the curve of the time series in that bin, as well as the slope of the regression line of that curve.

Once we have an adequate representation for a single swipe, the next step is to induce a generalized model that represents the class of swipes generated by the user. It is important to ensure that we have *reliable*, *consistent* swipe data prior to inducing the first model; learning cannot take place if the data is extremely variable. The initial training set is composed of the first n swipes the user generates from the moment they first start using the application. The $(n+1)^{th}$ swipe is authenticated based on the

model induced from the first n swipes. Conforming to the rolling window framework outlined above, if the $(n+1)^{th}$ swipe is correctly authenticated, the training set is updated, and is now comprised of n swipes, from swipe number 2 to swipe $(n+1)$. In the eventuality that the swipe is rejected, the training set stays the same.

Model induction proceeds as follows: we begin by first generating an exemplar swipe from the current training set. Each swipe s_i in the training set is represented by a vector $(touch_i, accel_i, gyro_i)$, where $touch_i$, $accel_i$ and $gyro_i$ are the vectors representing the touch, accelerometer and gyroscope features respectively (calculated as described previously). The exemplar vector is simply the *mean vector* for the window, $swipe_{mean}$, which is a vector composed of the feature means of the swipes in the window.

Once we have the mean, we calculate the euclidean distances of each swipe s_i in the training set from $swipe_{mean}$; these distances are calculated separately for each sensor. This separation allows us to weigh each sensor independently of the other sensors.

Once we have the distances, we compute the final value v_i for $swipe_i$ as being the product of the sensor distances: $v_i = touch_{distance} \times accel_{distance} \times gyro_{distance}$. Authentication of a new swipe is conducted by employing these distances and $swipe_{mean}$. Let the new swipe be s_{new}; we calculate the value v_{new} between s_{new} and $swipe_{mean}$ as described. We also compute μ_v and σ_v, the mean and standard deviation of the values for the swipes in the window. If $v_{new} \leq (\mu_v + k\sigma_v)$, where k is a user specified constant indicating how many standard deviations from the mean legitimate swipes are allowed to be, we authenticate the swipe as being legitimate and add it to the window. Given the domain, it is imperative for the classifier to be computationally efficient, as it is to run on users mobile devices.

The reader may note that we do not employ the classifiers described in the previous chapters. The primary reason for this is that both the autoassociator (AA) and the one-class support vector machine (ocSVM) have a significant number of parameters that require to be tuned for a particular dataset. In our case, each users data represents a unique dataset, and consequently, each classifier will likely require

a unique set of parameters. Given that usability is highly important, it is not practical to put the onus of selecting the optimal set of parameters on the user (especially given that the vast majority will not be skilled in machine learning). Thus, we employ the simple classifier described in this section. In the following section, we describe the experimental framework employed to test the performance of this classifier over different user motions.

5.5 Experimental Framework

As discussed, we identify two behaviours on which to split the swipe space: sitting and walking. While it is possible to identify a lot more behaviours, as an initial step, we focused only on these two. The user was given the application installed on the phone and requested to use the app while sitting using a unique profile name. Once this was accomplished, a new profile was created and the user was asked to only swipe while walking. In other words, we obtained two data sets, one representing walking swipes and one for sitting. Identifying whether a user is stationary or in motion is simple; modern phone operating systems have API's that are able to detect motion. For example, one can employ the *step detector* capability in the Android API. Thus, when a user swipes, the phone can detect motion, or the lack of, and can process the swipe using the appropriate one-class classifier. As a rseult, the learning framework we employ is that *full knolwedge*, since we can identify the which sub-concept novel instances belong to.

Two users we asked to generate these datasets, and the results are presented over these. In particular, the mixed motion results (*i.e.* over the whole domain) are are obtained by combining all the datasets and simulating them through our algorithm. The results by focusing on the sub-concepts are obtained by simulating the datasets for sitting and walking separately, and combining the results for the resulting ensemble. Finally, we only present the results by ignoring the touch screen data. Given that our focus is on analyzing under user motion, we only consider the sensors that are affected by motion, namely the accelerometer and the gyroscope.

Thus, we can treat these results as being for the perfect imitation attacks with respect to the touch patterns.

The product was still at an early developmental stage, and thus, its use and exposure outside the company was heavily restricted. As a result, the experiments were conducted in-house with two employees. Subsequent to the company officially deploying the first prototype of the product to the public, we aim to conduct experiments with more users.

The first employee, User 1, swipes with one hand, his right, while holding the phone in the same hand. His swipes are highly consistent with respect to the touch patterns. The second employee, User 2, holds the phone in both hands, and swipes with the index finger as well as the thumb using her right hand. Her swipes are less consistent than those of user 1. The phone employed in both cases is the HTC One.

We discuss two sets of results. In the first, we test the ability of users to authenticate themselves (the User Acceptance Rate). In the second set, we test the ability of our solution to defend the user against *attacks* (the Attack Acceptance Rate); the user hands the phone over to another user and they attempt to authenticate themselves. These are presented in the following section.

5.6 Results

The results from our user study experiment are presented in Tables 5.2 and 5.3. Specifically, these numbers represent the number of times the user was able to log into his phone by using their swipe without being rejected, *i.e.*, the user acceptance rate (UAR), and the attacker acceptance rate (AAR), the rate at which the model authenticates attack attempts. It is worthwhile to note that typically, industry reports accuracy taken over three consecutive attempts; if even one out of these three attempts is successful, the entire set is labelled as a successful attempt. We, however, only consider single attempts. Thus, our numbers, compared to industry, represent a lower bound on performance, as taking sets of three will increase the accuracy values. We report UARs and AARs separately for each motion for clarity. The results for

Table 5.2: Authentication accuracies for different motions for User 1.

Motion	User Acceptance Rate	Attacker Acceptance Rate	g-mean
Sitting	68.42 %	29.49 %	.694
Walking	61.9 %	16.93 %	.717
Ensemble	65.6 %	23.8 %	.707
Mixed	55 %	39.2 %	.578

Table 5.3: Authentication accuracies for different motions for User 2.

Motion	User Acceptance Rate	Attacker Acceptance Rate	g-mean
Sitting	71.42 %	12.5%	.79
Walking	54 %	47.36 %	.533
Ensemble	63.7 %	27.9 %	.677
Mixed	63.2 %	73.6 %	.408

the system over both sub-concepts are shown under *Ensemble*, whereas the results for the model built by ignoring motion are shown under *Mixed*. Finally, we also present the g-mean for both systems.

5.6.1 Discussion

If one were to observe the raw time series generated by different motions for the sensors, it will be clear that even though the user is the same, each motion corresponds to a unique sub-concept. This is illustrated, for sitting and walking for the user, in Figure 5.11 and Figure 5.12 for a single swipes x-axis accelerometer and gyroscope, respectively.

As has been the case in the domains described before, the presence of multiple sub-concepts makes it very difficult to create one-class classifiers that work well over all. This is evident by the results presented here; attempting to model both motions results in poor performance due to the model over-generalizing over the motions. This is in line with the analysis in the previous chapter, where we discovered that complex domains will result in classifiers that over-generalize, resulting in poor performance. This is particularly significant for User 2, who is the more variable of the two users; the gains in performance are exceptionally high with respect to the AAR.

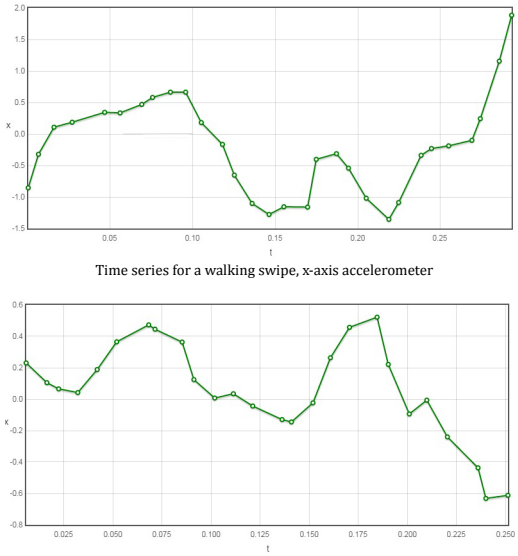


Figure 5.11: Accelerometer time series for walking and sitting.

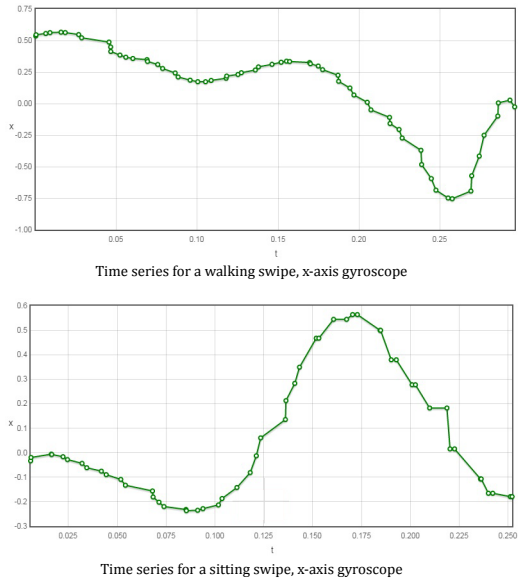


Figure 5.12: Gyroscope time series for sitting and walking.

5.7 Conclusions

The ubiquity of smartphones in our daily lives has greatly accelerated research into the application of such biometrics into the devices for providing security. In this chapter, we introduce our work in employing users swipes as a behavioural biometric for authentication on mobile devices. This domain is a paragon for the properties being considered in this thesis; the only data typically available is that from the user, and given that a users motion impacts the readings of the sensors recording the swipe, the underlying distribution is inherently complex.

To this end, we identified walking and sitting as the two most general sub-concepts and proceeded to design the swipe authentication system within the framework described in this thesis. Specifically, since user motion can be identified explicitly on smartphones, we employ the *full knowledge* framework. We installed the application to record swipe data on an Android phone and requested a user to use it while sitting and while walking, thus creating two distinct profiles. While this may not simulate usage behaviour in the real-world, this framework is a close enough approximation and allows us the flexibility to control the data collection process. The data from the two profiles are simulated through our authentication process separately and as a whole (the latter representing the mixed behaviour). The resulting UARs, AARs and the g-mean are significantly better for the system employing full domain knowledge of sub-concepts.

The swipe-authentication domain presents a highly challenging and complex learning task; the technical sophistication in modern smartphones allows us the luxury to explicitly identify the sub-concepts inherent in the domain. However, as we have pointed out, this may not always be the case. In the next chapter, we look at the domain of anomaly detection in gamma-ray spectra, where identifying the sub-concepts can only occur during training. This domain, thus, is tested against our second framework, where we employ fuzzy knowledge.

Chapter 6

Case study - gamma-ray spectra data

In this chapter, we elaborate our research, in conjunction with Health Canada, in the construction of a one-class classification framework in the domain of invasive isotope detection over gamma-ray spectra. More specifically, physicists at Health Canada were interested in automating the identification of concealed radioactive materials that have a potential to pose a significant threat to attendees at large public gatherings. This ability is of specific importance as it reduces the reliance on human experts and presents the possibility of increased accuracy, and our preliminary work on this is presented in Sharma *et. al* [71].

While the previous chapter considered one-class classification when we have knowledge of, and can detect, the sub-concepts inherent in the domain, in the domain considered in this chapter, while we do have knowledge regarding the sub-concepts, detecting them is not possible. Thus, the framework employed for boosting the performance of one-class classifiers is that of fuzzy knowledge; sub-concept detection is performed by the inducing a multi-class classifier.

We begin this chapter by describing and analyzing the domain in Section 6.1. Section 6.2 then describes our proposed learning system in detail, followed by an explanation of the experimental framework employed in Section 6.3. The results from both the original system employed by Health Canada and our two-tiered system

are presented and discussed in Section 6.3. We summarize our findings and insight in Section 6.5.

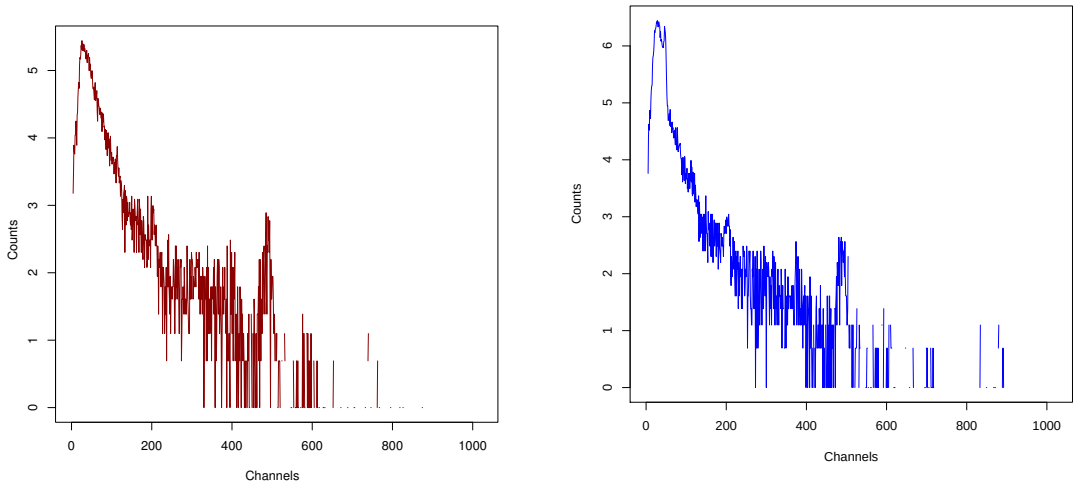
6.1 Gamma-Ray Spectrometer Data

Health Canada deployed eighteen NaI (Sodium Iodide) detectors during the Vancouver Winter Olympic Games in 2010 in order to produce a catalogue dataset that can be used for future development and testing of Machine Learning approaches to multi-category alarm systems. A gamma-ray spectrometer is an instrument that measures the distribution of the intensity of gamma radiation versus the energy of each photon. The data was collected in one-minute intervals during the games, thereby producing a dataset of 708,480 spectra readings. The resulting measurements can be plotted as in Fig. 6.1 *(i)* and Fig. 6.1 *(ii)*; the former corresponds to a pure background measurement, whereas the latter corresponds to a background plus Technetium. This realization of the data is in log form, which is the standard method for viewing and analyzing such data. Energy is represented in terms of channels on the x-axis and the counts, which indicate the intensity, are recorded on the y-axis. The lower channels in the plots correspond with lower energy photons; the energy increases with the channels. The isotopes of interest in our experiments all peak well below channel 600, and thus, to minimize the effects of the so-called 'curse-of-dimensionality', we only utilize the first 600 channels.

In the following sub-sections, we will analyze the data in terms of imbalance and distributional complexity.

6.1.1 Learning Complexity

We were provided with data from three stations, and apart from the background, the readings contained spectra for three medical isotopes, namely Iodine, Thallium and Technetium (one of the stations also had readings for Caesium, which were the result of a check-source). The resulting datasets displayed a prohibitive level of imbalance, as evident from the number of isotope spectra in the data from each station. These,



(i) (ii)

Figure 6.1: These figures contain randomly selected examples from the gamma-ray spectra data set. Plotted on the log-scale, sub-figure *(i)* depicts a background instance and sub-figure *(ii)* depicts and instance containing the medical isotope Technetium.

along with the number of background instances, are enumerated below:

- Station 6: 5 Iodine spectra, 3 Technetium spectra and 15 Ceasium spectra, 39,000 background spectra.
- Station 12: 2 Iodine spectra, 7 Technetium spectra, 25,747 background spectra.
- Station 13: 3 Iodine spectra, 2 Technetium spectra, 2 Thallium spectra, 24,709 spectra.

The large amount of background spectra, which, coupled with the lack of data from medical isotopes, made the application of any multi-class classification system a futile endeavour. The imbalance ratios for Stations 6, 12 and 13 are $1 : 1696$, $1 : 2861$ and $1 : 3673$, respectively. Given this severe level of imbalance, based on our analysis from Chapter 3, one-class classification is the only viable option. Specifically, we perform anomaly detection by learning a one-class classifier over the background data.

In addition to the medical isotopes that were measured and identified as a result of people passing by the Health Canada detectors, Health Canada also provided

artificially generated spectra for Cobalt at varying signal strengths. These were subsequently incorporated into the data from the receptors for all stations. The sole purpose of this data was to facilitate proper evaluation; the lack of medical isotope data, while hindering the training of binary classifiers, also poses an issue for evaluation. Thus, in order to accurately verify the strength of the final system, these instances were employed during evaluation.

6.1.2 Domain Complexity

The physicists at Health Canada indicated that the presence of rain and/or water had an impact over the gamma-ray spectra measured. In this subsection, we highlight the distributional distinctions between the background and medical isotope instances when they are, and are not, affected by rain. To being, consider Figures 6.2, 6.3 and 6.4. These display the mean mean and non-rain spectra for Stations 6, 12 and 13, respectively. Simple visual inspection highlights that there is a small but noteworthy difference between rain and non-rain spectra at the lower energy levels. In particular, the most obvious distinction is the "hump" over channel 200; this is a direct consequence of water affecting Radon signatures in the background.

While the difference appears minimal, in practice the effect of rain can have the potential to impact and complicate the classification process. Let us examine the PCA plots for the first three principal components for all three stations. These are provided in Figures 6.5, 6.6 and 6.7. While the previous plots illustrated the point that there is a difference, these highlight the impact that difference can have on a classification system. We observe that the rain concept has a much wider spread than the non-rain concept. As was the case in the domain of user swipes, we have a tighter sub-concept within a wider concept. As a result, learning a single one-class classifier will result in over-generalization; a greater number of dangerous isotopes will pass by undetected (high number of false positives).

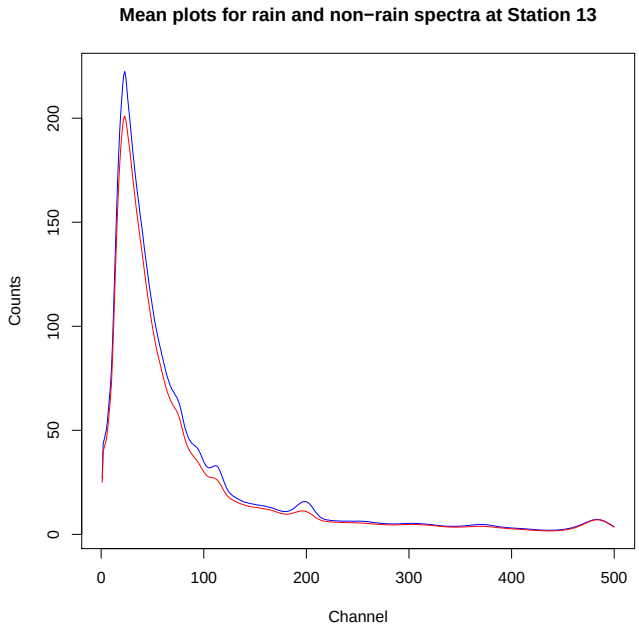


Figure 6.2: Mean rain and non-rain spectra for Station 6. Blue represents rain and red non-rain.

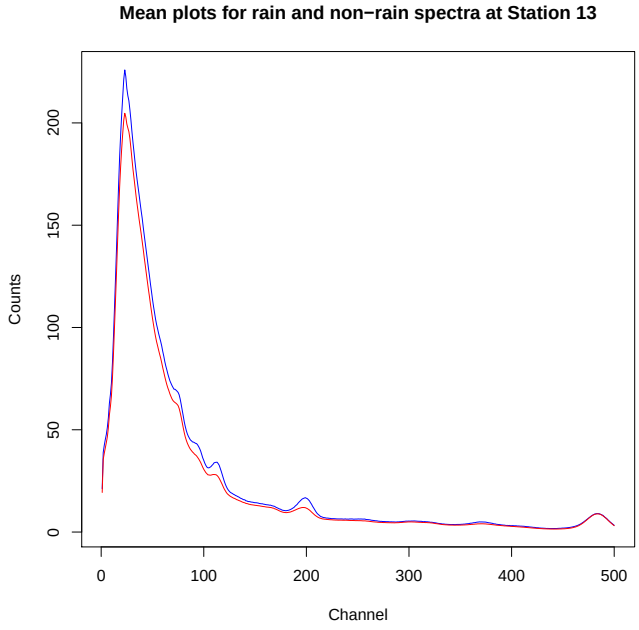


Figure 6.3: Mean rain and non-rain spectra for Station 12. Blue represents rain and red non-rain.

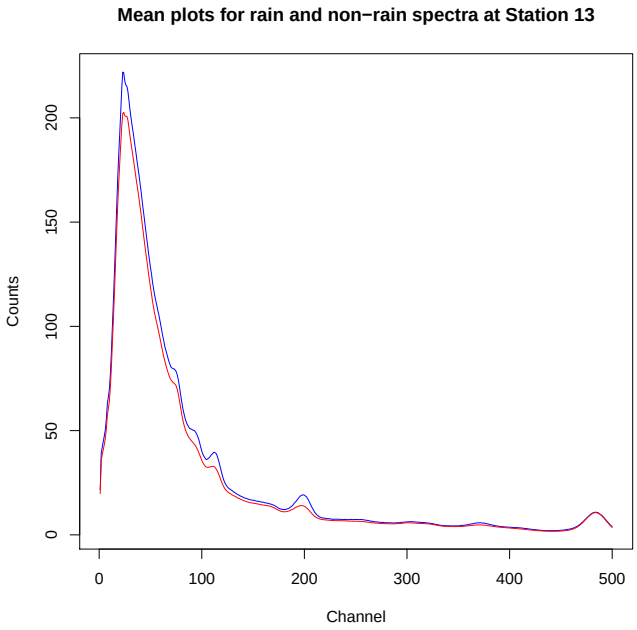


Figure 6.4: Mean rain and non-rain spectra for Station 13. Blue represents rain and red non-rain.

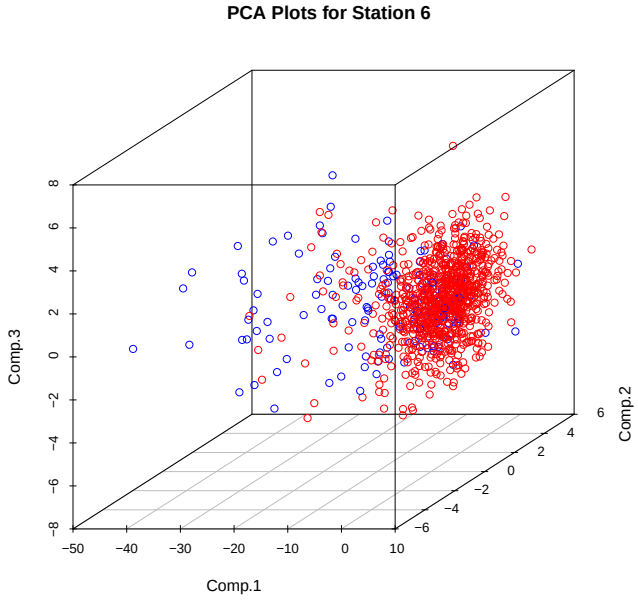


Figure 6.5: The first three principal components for Station 6. Red represents no rain and blue represents rain.

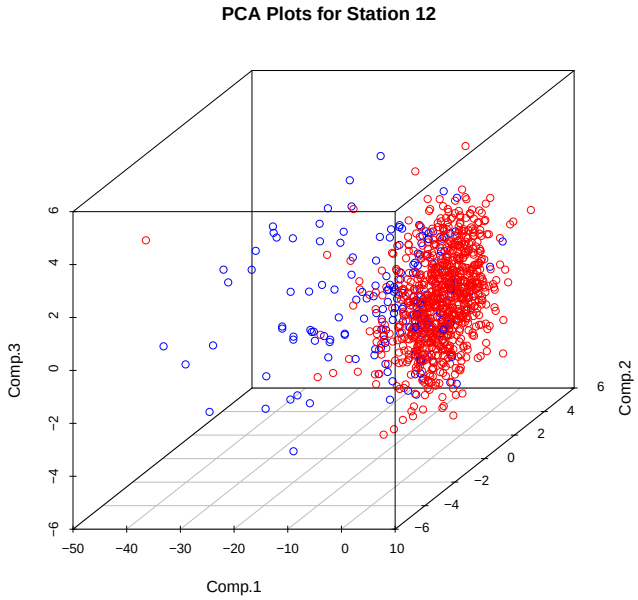


Figure 6.6: The first three principal components for Station 12. Red represents no rain and blue represents rain.

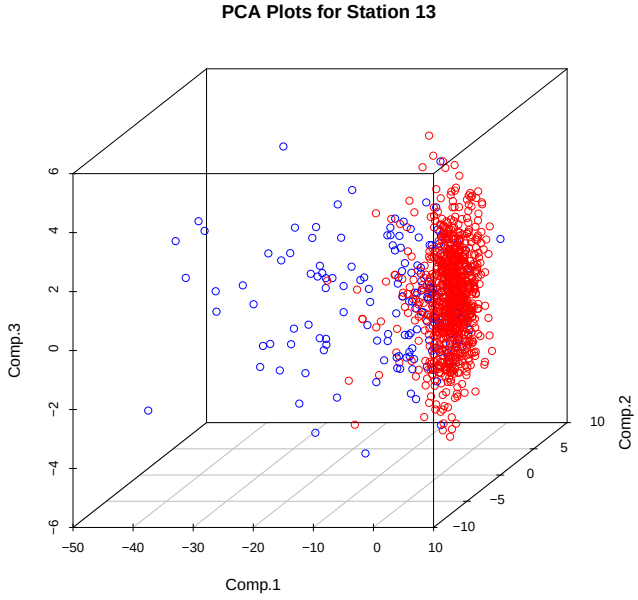


Figure 6.7: The first three principal components for Station 13. Red represents no rain and blue represents rain.

6.1.3 Summary

The previous sections examined the domain of gamma-ray spectra from the two perspectives laid out in this thesis. We first considered the level of imbalance. All three stations exhibited imbalances so severe that the only learning paradigm that can be applied is that of one-class classification. Secondly, we examined the complexity of the data space. In particular, the non-rain data was present as a tight sub-concept within the rain data. This is to be expected as the presence of water causes spikes in the spectra, resulting in a wider distribution. Thus, it is prudent to explicitly separate these two concepts and build a classification system within the confines of the framework presented in this thesis. This will be elaborated in subsequent sections.

6.2 Learning Framework

In this section, we discuss the system that has been devised to detect isotopes of interest in the Health Canada dataset. As the analysis in the previous section demonstrated, the data comprises of two distinct sub-concepts, based on whether they are affected by rain/water. While we have *a priori* knowledge of these during training, it is not possible to ascertain, during classification, which sub-concept the novel spectrum will belong to. This is because it is not just rain that affects a spectrum; the presence of water in the environment has an impact as well. Thus, the learning framework we employ is that of *fuzzy knowledge*; we train a binary classifier to learn to discriminate between the two sub-concepts, and pass novel spectrum to the appropriate one-class classifier. Fig. 6.8 depicts the architecture of the ensemble classification strategy.

In subsequent sections, we detail each learning phase.

6.2.1 Phase I: Rain separation

In this phase, we rely on the availability of labelled training data to build a classifier that can recognize the presence of a heavy rain event. We are specifically interested

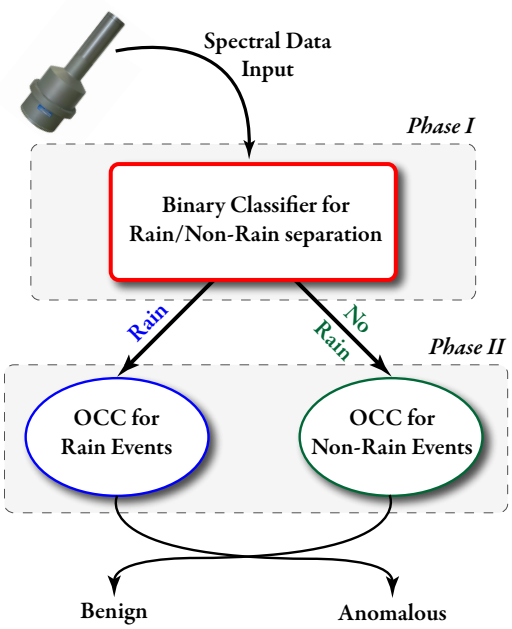


Figure 6.8: The proposed architecture for anomaly detection in gamma-rays.

in heavy rain events as they perturb the background distribution in such a way that the identification of spectral events of interest becomes significantly more difficult.

Unlike in the previous domain of mobile security, where a phones' sensor can tell concretely the sub-concept to which the novel instance belongs, in this domain, it is impossible to tell *a priori* whether a spectra being measured belongs to a rain event. However, a variety of learning strategies may be appropriate for this phase of the system. In theory, there is sufficient rain data in the dataset to apply binary learning. This is ideal, as binary classifiers are known to produce very strong results. As a result, we have applied a set of binary learners to this problem; the specifics of which are discussed in the following section.

An important consideration while building discriminatory or recognition-based models is whether or not to include medical isotopes into the training sets. When considering rain and non-rain as separate classes, the dichotomies generated based on these classes themselves constitute of two classes, namely background and isotopes. Conversely, when considering background and isotopes as separate classes, the dichotomies contain rain and non-rain as the two sub-groups. Since, in the first

tier, our aim is to treat rain and non-rain as the two distinguishing classes, in order to induce a classifier as general as possible, it is imperative to consider the signals generated by both background and isotope spectra during rain and non-rain events, as removal of either of the two will lead to a loss of information, thus impacting the generalization capabilities of the classifier. This conforms to our analysis in Chapter 4 regarding implicit concepts as containing both targets and outliers. As a result, in our experiments, during the first tier, we include both background and isotope spectra during training.

6.2.2 Phase II: Isotope/Anomaly Detection

Subsequent to rain classification, the instance is passed to the appropriate rain or non-rain model, where it is assessed for its normality. Specifically, is this another background instance, or is it deserving of further consideration.

Pursuant to our previous discussion regarding the imbalance resulting from the scarcity of spectra of isotopes of interest, this phase relies solely on an anomaly detection algorithm. Our initial experiments on obtaining rankings from each of these algorithms have led us to conclude that the Mahalanobis Distance (MD) offers the best results, as opposed to either the Autoassociator (AA), support vector machine (SVM) or the Variance in Angle Spectrum (VAS). Both the AA and VAS have extremely long training times, which increase drastically proportional to the dimensionality of the data being trained. But perhaps crucially the greatest advantage that the MD has over AA and VAS is the near-Gaussian nature of the spectral data. MD calculates the distance explicitly assuming a Gaussian distribution; given the nature of the data provided, this allows MD to be remarkably accurate, thereby giving rankings that are as close to being probabilistically accurate. It is for this reason that we chose the MD over the others for the second phase of our system.

6.2.3 Threshold Selection

Selecting an appropriate threshold is critical to turn our system from one that produces rankings to one that produces a classification, and consequently, an alarm. The ROC curve offers an intuitive manner by which to calculate an appropriate threshold by employing the true and false positive rates attained by every point on it. Two popular approaches are enumerated below:

1. *Youden Index*: This measure is based on maximizing the sum of sensitivity and specificity, based on the intuition that a good threshold will produce a large sensitivity and specificity. It assumes that both values are equally important. Intuitively, this corresponds to the point farthest from the diagonal in the ROC space.

2. *Closest point to (0,1)*: The perfect ROC curve would pass through the point (0,1) (sensitivity and specificity are both 1 for a threshold). Therefore, one way to select a threshold is to select the point closest to (0,1) on the ROC curve.

Both Youden index and the Zero-One point rely on the ROC curve. However, by the nature of the geometry by which the thresholds are calculated, the Youden index offers a higher false positive rate as compared to the zero-one, the trade-off being a lower true positive rate for the zero-one threshold. As we are more interested in reducing the number of false positives, we rely on the zero-one index in our experiments.

6.3 Experimental Framework

This section details the experimental framework undertaken to validate the performance of our two-tier system for anomaly detection. We begin by providing a detailed description of data pre-processing, followed by the training and testing procedures used for validating the system.

6.3.1 Data Pre-processing

The initial phases of pre-processing have been described in Section 6.1. Data from Stations 6, 12 and 13 was split into two parts, namely background data and medical isotope data. Health Canada had also provided us with artificially generated Cobalt data, which was divided into two parts, rain and non-rain. Each of these contained Cobalt spectra of varying intensities, ranging from 50 (the weakest strength) to 500 (the strongest strength). In accordance with our discussions with the experts at Health Canada, we augmented this data with the medical isotope data from each of the three stations; it was made clear to us that the Cobalt data was station independent, and thus could be used in conjunction with the medical isotopes from all stations.

The issue of high dimensionality was resolved by taking only the first 600 channels from the 1024 channel spectrum. During the course of our meetings with the experts from Health Canada, we were made to understand that signals of interest would not appear beyond channel 600, and therefore, we could restrict our data to only the first 600 channels. Thus, in all our experiments, we utilize only the first 600 channels.

The background data from each station is further divided into two sets, one for rain events and one for non-rain events. The labelling of the background data into rain and non-rain events is done based on information provided by Health Canada, and the weather information for Vancouver provided by Environment Canada. The purpose of this explicit labelling is to facilitate binary learning for Phase I. Both sets are combined, and divided explicily to facilitate 10-fold cross validation. The validation set from the background data is augmented with the medical isotope data to produce our final testing set.

6.3.2 Training and Testing Procedures

As detailed in previous sections, our system is a two-tiered model. Phase I is comprised of a binary learning algorithm, which aims to induce a discriminant function that can separate normal spectra from spectra observed during heavy rain events. We

use three different binary learning algorithms, namely Decision Trees, Naïve Bayes and an Instance Based Learner. The training set extracted from the background data, as described in the aforementioned sub-section, is passed on to the binary learner of Phase I. Once the classifier has been induced, we can label the testing data, comprised of the remainder of the background data along with the medical isotope data, and proceed to Phase II.

Phase II consists of two anomaly detection systems, one for rain events and another for non-rain events. Specifically, we use the Mahalanobis Distance (MD). The Mahalanobis distance relies on the calculation of a mean and a covariance matrix. These are calculated using the training data; the matrices for the rain system are calculated using the rain events from the background training set, and conversely, the matrices for the non-rain system are calculated using the non-rain events from the background training data. Note that we only use the background data for calculating the matrices, since the anomalies we expect to find are relative to the background.

Once we have our matrices, the rain and non-rain datasets of the testing data resulting from Phase I are passed to the appropriate anomaly detection system, and once the MD has been calculated, we obtain our final rankings. It is expected that the spectra displaying the greatest anomaly to the mean would have the largest distances, and vice versa.

Fig. 6.9 graphically illustrates the entire experimental framework. The following section will present the results obtained based on this framework.

6.4 Results

The results obtained from the experiments conducted, as described in the previous section, are presented here. We first produce the Area Under the ROC Curve (AUC) values for each of the three binary classifiers, for each station, followed by the ROC plots. Furthermore, in order to verify whether the classifier induced during Phase I does indeed provide an improvement over the final rankings obtained from Phase II, we also present the AUC values obtained in the absence of Phase I. Specifically,

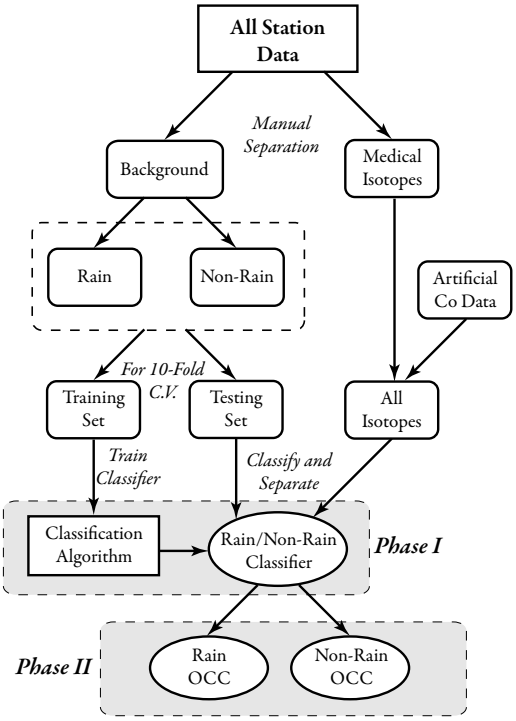


Figure 6.9: This illustrates the complete experimental framework undertaken to validate our proposed architecture.

we build the matrices for MD using both rain and non-rain background data, and produce distance rankings using a testing set comprised of both rain and non-rain background and medical isotope data. We conclude the section with a discussion elaborating on the conclusions derived from these results.

6.4.1 Results from the Two-Tier Anomaly Detection System

Table 6.1 presents the AUC values obtained from the rankings produced by the system over all stations, using each of the three binary classifiers, during Phase I.

The ROC curve for the results obtained by Health Canada for stations 6, 12 and 13 are given in Figures 6.11 and 6.10. For our architecture, in all cases, the Naïve Bayes Classifier (NBC) produces the best results, and thus, we display the associated ROC curves for NBC at stations 6, 12 and 13, for both rain and non-rain events in Figures. 6.12, 6.13 and 6.14. These serve to illustrate that employing machine learning results in a marked improvement in performance.

Table 6.1: AUC values for all Stations, over all binary classifiers, for both Rain and Non-Rain anomaly detection systems.

Results over Rain Classifications from Phase I								
Station 6			*Station 12*			*Station 13*		
IBK	J48	NBC	IBK	J48	NBC	IBK	J48	NBC
0.993	0.991	0.994	0.998	0.998	0.998	0.999	0.999	0.999
Results over Non-Rain Classifications from Phase I								
Station 6			*Station 12*			*Station 13*		
IBK	J48	NBC	IBK	J48	NBC	IBK	J48	NBC
0.992	0.993	0.994	0.999	0.999	1.000	1.000	0.999	1.000

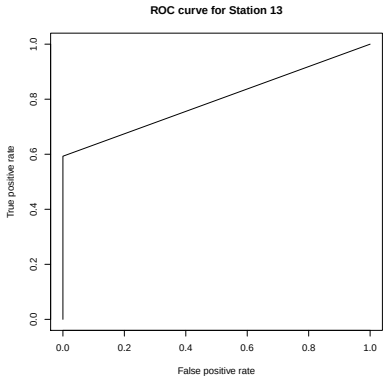
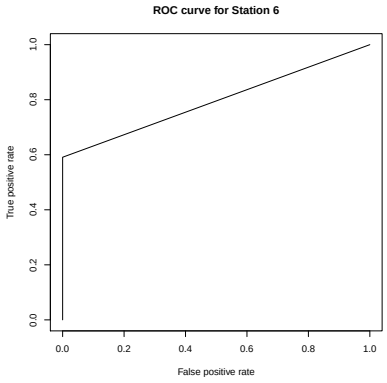
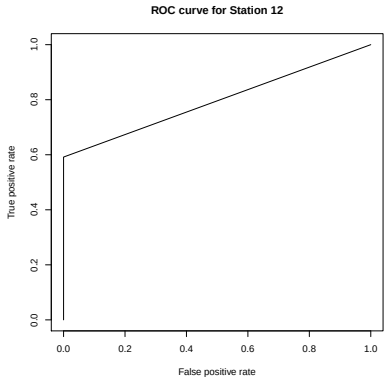


Figure 6.10: The ROC plot for Station 13, generated using the results obtained by utilizing the techniques employed at Health Canada.

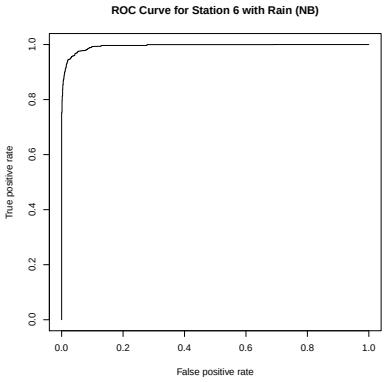


(i)

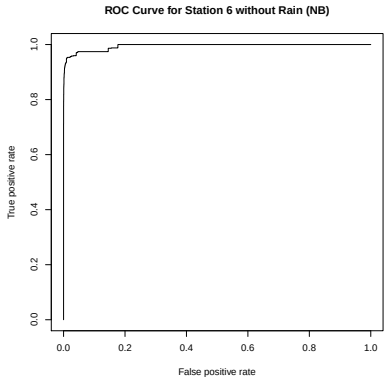


(ii)

Figure 6.11: The ROC plot for Station 6 and Station 12, generated using the results obtained by utilizing the techniques employed at Health Canada.

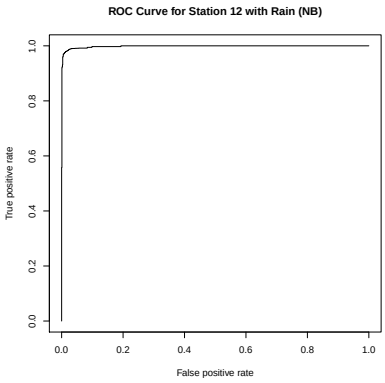


(i)

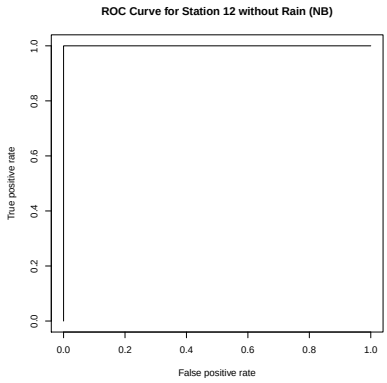


(ii)

Figure 6.12: The ROC curves for Station 6 for both rain and non-rain events, using the Naïve Bayes Classifier and the Mahalanobis Distance..

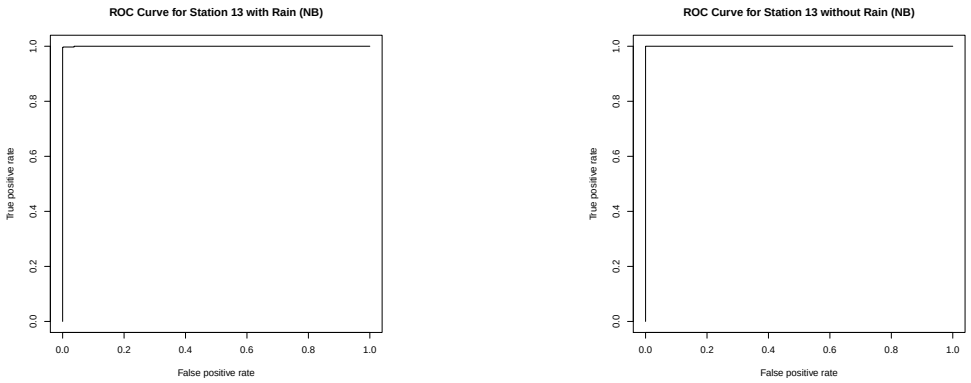


(i)



(ii)

Figure 6.13: The ROC curves for Station 12 for both rain and non-rain events, using the Naïve Bayes Classifier and the Mahalanobis Distance.



(i) (ii)

Figure 6.14: The ROC curves for Station 13 for both rain and non-rain events, using the Naïve Bayes Classifier and the Mahalanobis Distance..

Table 6.2: AUC Values from no rain separation.

Station	**AUC Value**
Station 6	0.991
Station 12	0.999
Station 13	0.999

6.4.2 Results from no rain separation

Table 6.2 shows the AUC values for the system when no binary classifier is induced in order to separate rain spectra from non-rain spectra; the classifier is trained over both sub-concepts.

6.4.3 Discussion

The AUC values for NBC in Table 6.1 demonstrate the spectacular performance of the two-tier system. Indeed, an AUC value of 1 is one that is the ultimate goal of every classifier; it implies a perfect ranking, devoid of any classification errors. This can be attributed primarily to the nature of the probabilistic distribution of the individual channels of the spectral data, a fact alluded to in Section 6.2.2. The near-Gaussian distribution of each channel makes the feature space of the domain exceptionally conducive to be used with the Mahalanobis Distance (MD). This in turn produces the results that we see here.

Table 6.3: Total number of True Positives (TPs) and False Positives (FPs) incurred with (2-Tier) and without (1-Tier) rain separation.

Station 6			
1-Tier		*2-Tier*	
TPs	FPs	TPs	FPs
3758	336	3950	144
Station 12			
1-Tier		*2-Tier*	
TPs	FPs	TPs	FPs
3856	98	3918	36
Station 13			
1-Tier		*2-Tier*	
TPs	FPs	TPs	FPs
3900	24	3914	10

It is interesting to note that even without using a binary classifier to split testing data into rain and non-rain classes, the MD based anomaly detection system still produces exceptional AUC values, as seen in Table 6.2. In comparison with results from Table 6.1, however, we note that the two-tier system outperforms the single-tiered system over all stations. Even though the differences in the results may appear to be minuscule, it is prudent to be wary of the fact that a minute difference in AUC values can still imply a large difference in correct classifications. In other words, the single-tiered system ranks more background instances over medical isotopes, and vice versa, as compared to the two-tier system. This is illustrated in Table 6.3, where we can see that without inducing a rain separating classifier, the single-tier system gives more than twice as many false alarms as the two-tier system. Note that the true positive and false positive values are given for the Naïve Bayes classifier based two-tier system, since this classifier provided the best AUC values overall. We also note that the improvement in performance is resulting, once again, in a reduction in the number of false positives due to there being less over-generalization.

6.5 Conclusions

The primary objective of the project undertaken was to devise an automated system that could identify and alarm appropriately on anomalous gamma-ray spectral readings, with little, if any, intervention on part of the human personnel involved. Machine Learning tools were considered in order to achieve the stated objectives, and the result was a two-tier one-class classification system for spectral anomaly detection. The domain presented to us fit the characteristics outlined previously in the thesis: it exhibited extreme levels of imbalance and had a complex underlying distribution. Based on data analysis, we attributed this complexity to environmental factors, particularly rain. The two-tier design was the devised based on the observation that spectral readings acquired during heavy rain events had been identified as being detrimental to effective anomaly detection, and therefore would need to be filtered out *a priori*. We were fortunate enough to have *a prior* knowledge as to during which days and time the recorded spectra would be affected. Thus, rather than blindly cluster the data for simplification, we were able to divide the data explicitly based on expert knowledge.

In Phase I of the two-tier system, we induce a binary classifier that learns to discriminate between spectral readings occurring during heavy rain events, and readings occurring during all other conditions. Spectra passed to this classifier would be classified into either case, and passed on to the appropriate anomaly detection systems (*i.e.* one for either rain or non-rain events), which comprise Phase II of the two-tier design. The Mahalanobis Distance is used as the anomaly detector in this phase, as the Gaussian assumption inherent in its calculations conform exceedingly well with the probability distributions of the individual energy channels of the data. The final product of the system is a ranking of spectral readings based on their Mahalanobis Distance from the appropriate background means, with higher values indicating a larger anomalous nature.

AUC values obtained by using the Naïve Bayes classifier during Phase I indicate that an overwhelming majority of medical isotope spectra are ranked above the back-

ground spectra. Of particular importance is the fact that the spectra for Cobalt, specifically spectra with weak signal strengths, are also ranked above the background spectra. This is a vast improvement over the systems employed at Health Canada, as we were informed that their systems could not identify spectra with strengths below 200 (the strengths ranged from 500, being the strongest, to 50, being the weakest). This further attests to the strength and accuracy of our anomaly detection system.

To conclude, we have developed a two-tier system for ranking spectra based on how anomalous they are from the background spectra. The two-tier design is a consequence of not possessing the ability to identify *a priori* the concept to which a novel instance belong. In the domain considered in the previous chapter, a mobile device's hardware made it possible to identify if a swipe was performed while the user was in motion or not. However, the hardware employed by Health Canada does not have the ability to detect whether water is affecting the isotope signatures. Thus, we employ binary classifiers to learn to discriminate between concepts of interest and perform this identification for novel instances. The success of this novel architecture is promising, as it demonstrates that in the absence of the ability to explicitly identify concepts in a one-class classification system, binary/multi-class classifiers can be employed to perform this identification implicitly.

Thus far, we have considered real-world domains in which, with the assistance of domain experts, we are able to ascertain sub-concepts within the domain, and separate data prior to learning in order to construct more efficient one-class classification systems. However, in the worst case scenario, such knowledge of sub-concepts might be unavailable. In the next chapter, we consider a domain that exhibits this property.

Chapter 7

Case Study - The CTBT Domain

The final domain we consider in this thesis pertains to data relating to the compliance verification of the Comprehensive Nuclear-Test-Ban Treaty (CTBT). The CTBT domain was originally introduced to the Machine Learning (ML) community in the form of an open data mining competition at ICDM 2008 [75]. The competition invited teams to take part in building classification models from a training set that was provided by the Radiation Protection Bureau of Health Canada (HC).

In the previous domains considered in this thesis, we had *a priori* knowledge as to the sub-concepts present within them; motion for the mobile security domain and rain for the gamma-ray spectra domain. However, in the CTBT domain, we do not have the luxury of such knowledge; the complexity of the domain, as we will elaborate further, is evinced by the poor performance of one-class classifiers over it. Thus, this domain presents a general, worst-case scenario for building one-class classifiers for complex distributions. The solution we employ still stays true to the core idea, *i.e.*, simplify the distribution based on similar sub-concepts. However, rather than employing explicit knowledge of these sub-concepts, which are not present, we cluster the domain; the sub-division is thus done based in a purely mathematical sense, with the underlying assumption that instances grouped by a clustering-specific similarity measure will still, in some way, correspond to a real world, domain-specific concept.

The remainder of the chapter is organized as follows. Section 7.1 introduces the domain, and the experimental framework is described in Section 7.2. The results are

presented in discussed in Section 7.3, followed by concluding remarks in Section 7.4.

7.1 Domain Description

The Comprehensive Nuclear-Test-Ban Treaty (CTBT) is a multilateral treaty that was adopted on September 10, 1996 by the United Nations General Assembly. The goal of the treaty to to prevent the proliferation of nuclear weapons; all signatory states agree to a ban on nuclear explosions for military or civilian purposes. For compliance verification, a monitoring network of 337 facilities exists around the world.

Amongst the technologies used for compliance verification, namely hydro acoustic, infrasound, seismic and radionuclide monitoring [77], the latter provides the only means for unambiguously discriminating a low-yield, clandestine nuclear explosion from other, background events. Thus, in support of the CTBT, monitoring stations with the capability of sampling and measuring the active concentration of four radioxenon isotopes, namely ^{131m}Xe, ^{133}Xe, ^{133m}Xe, and ^{135}Xe by SPALAX technology [75, 21], have been installed at numerous sites across the globe.

7.1.1 Learning Complexity

It is easy to see why this domain would be extremely difficult to model as a binary classification problem. The stations are continuously monitoring and gathering data, resulting in an abundance of data pertaining to benign atmospheric readings. With respect to readings corresponding to nuclear tests, since the creation of the treat, only three countries have carried out nuclear tests; India and Pakistan in 1998, followed by three tests conducted by North Korea in 2006, 2009 and 2013. Thus, there is an immense imbalance between the the two classes of the domain.

For the competition, Health Canada provided with data from five geographically distinct locations; the data set was composed of measured concentrations of ^{131m}Xe, ^{133}Xe, ^{133m}Xe, and ^{135}Xe. However, since there was no available explosion data, they provided with synthesized explosion data. This allowed for the application of binary classifiers, and the the top results ranged from an AUC of 0.75 to 0.83, and applied a

series of sampling strategies in conjunction with binary classification techniques, such as SVM. Subsequent to the competition, HC continued to refine their data set, and explore the application of ML techniques for the verification of the CTBT. Stocki *et. al* [76] conducted further experiments on measured background concentrations and synthesized explosion data (due to the absence of any data from explosions), utilizing data from a single receptor site, collected from 1 June, 2004 to 30 June, 2005. The best classifier was determined to be the k-nearest neighbour algorithm, with an AUC of 0.988.

Due to the fact that the data was synthesised, there was enough explosion data to facilitate the induction of binary classifiers. However, it is not a realistic premise to assume that there will be enough explosion data to induce a binary classifier; nations do not typically conduct nuclear tests on a regular basis. Thus, explosions are, for all practical purposes, non-existent. Furthermore, due to the significant variability in the environment, the possible release scenarios and locations, the explosion-space is vast. Consequently, an impractical number of explosion instances would be required in order to accurately model the distribution, and this is unrealistic.

Bellinger *et. al* [6, 5, 7] noted the 'unnatural' *a priori* class probabilities that are inherent in the publicly available Health Canada CTBT dataset, highlighting that the domain clearly fits into a one-class classification problem. This motivated them to develop a Stochastically Episodic (SE) event modelling and simulation framework. The framework assists in modelling SE events as they are propagated through the background noise. Bellinger *et. al* [6] demonstrate how a SE event, such as the inadvertent release of radioxenon from a clandestine nuclear test, propagates and decays within the background noise, the noise being a result of industrial nuclear activities. It is data generated from this framework that we employ for our work. The background noise-like non-SE pollutants are modelled as Gaussian plumes, and SE contaminants as Gaussian puffs. Both of these Gaussian models have been extensively studied in the literature, (Arya *et. al* [2, 73] for example). They have also been broadly applied to the modelling of radionuclides and other pollutants in the atmosphere [3, 11, 62, 80, 81].

In the following section, we analyze the complexity of the dataset with respect to multi-modality and overlap.

7.1.2 Domain Complexity

Figure 7.1 shows the first three principal components of the CTBT data we utilize for our experiments. As we have no knowledge of regarding the sub-concepts, we analyze how the two classes exist as a whole within the domain.

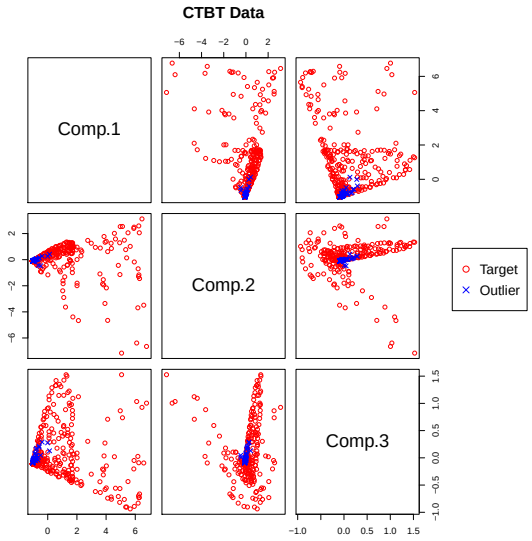


Figure 7.1: The first three principal components of the CTBT data.

While the dataset itself is not multi-modal, there is a high degree of overlap between the two classes. Indeed the outlier class appears to be almost completely contained within the target class. Furthermore, the target class, while having a dense cluster where the bulk of the data appears to exist, is highly spread. Target data that exists in the sparely populated regions could correspond to readings taken during abnormal atmospheric conditions.

7.1.3 Summary

The CTBT domain, as highlighted, presents an extreme imbalance, as data from the explosion class is, for obvious reasons, non-existent, making binary classification

impossible. Furthermore, while we cannot analyze how the various sub-concepts exist, visual analysis of the domain as a whole displays a high level over overlap between the two classes. Thus, based on our analysis thus far, the domain is indeed complex, utilizing the framework for *no knowledge* from Chapter 4 should improve the performance of one-class classifiers induced over this domain. These experiments and results are elaborated in following sections.

7.2 Experimental Framework

The framework employed is similar to the one from Chapter 4. We detail it here for completeness. The one-class classifiers employed are the autoassociater (AA) and the probability density estimator (PDEN). PDEN has also been implemented in WEKA [29], and we use the Gaussian Estimator as the density estimator, and AdaBoost with Decision Stumps as the class probability estimator.

The experiments with the AA were implemented using the AMORE R package, and run in R. One hidden layer was used for the AA in all the experiments, and the number of training iterations was set to 50. The momentum value was set to 0.99, and the learning rate to 0.01, and the number of hidden units ranged from 1 to the number of dimensions of the particular dataset. The number of clusters varied from 2 to 20. Once again, we use the geometric mean of the per-class accuracies as the evaluation metric. It is given by $g-mean = \sqrt{acc_1 \times acc_2}$, where acc_i is the accuracy of the classifier on instances belonging to class i. Evaluation is done using stratified 10-fold cross validation.

7.3 Results

The results of the OC classifiers over the CTBT dataset are presented in Figure 7.2. In particular, the dataset has 5000 normal instances, and 20 instances representing explosions.

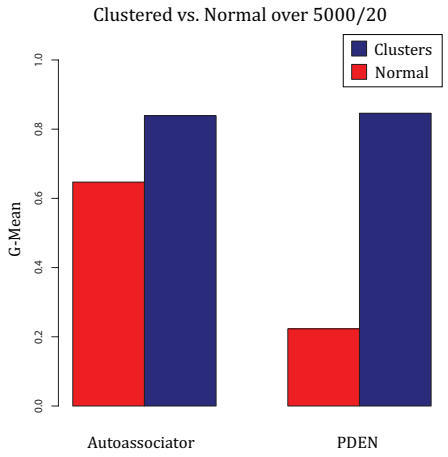


Figure 7.2: Results of the clustered and non-clustered (normal) autoassociator and PDEN over the CTBT dataset.

Discussion

Clustering the domain yields significant improvements in performance. This is particularly evident with the PDEN classifier. The visual analysis we presented earlier in the chapter makes this result unsurprising; the outlier class is almost entirely overlapped by the target class, and the latter exhibits a vast spread across the data space. Clustering allows us to divide the target class into smaller sub-spaces (with the hope of capturing the sub-concepts). Thus, the one-class classifier induced in the cluster that contains the bulk of the outlier data will contribute to an improved overall classification, as the classifier learnt over it will be far more specialized (*less generalized)* than a single classifier trained over the entire target class.

7.4 Conclusions

More often than not, we have enough knowledge regarding the domain that we can simplify the domain explicitly by dividing the space. However, there may be cases when such knowledge may not be present, or it may not fuzzy. In such situations, even trivial division by employing clustering can ameliorate one-class classifier performance. The reason stems from the analysis presented in Chapter 4; the key is to create sub-divisions such that the instances within each division are related to

the other instances, either through a concrete concept, or by a simple mathematical metric.

We have examined three real-world problems, all exhibiting the properties considered in this thesis. In all cases, our framework has improved the performance of one-class classifiers. In the following chapter, we conclude our thesis with summarizing our research and proposing avenues for future research.

Chapter 8

Concluding Remarks

This thesis introduces a general one-class classification framework that learns within the context of domain concepts. Data typically exhibits extreme levels of imbalance; classes of interest are exceptionally rare or non-existent and this renders the application of binary classifiers problematic. The data that is available tends to be derived from highly complex distributions, and a single concept may indeed contain multiple, distinct sub-concepts. This further complicates model induction, especially from a one-class classification context. Furthermore, each domain has its own nuances and quirks which, if ignored, can lead to inefficient classification systems.

The ubiquity and power of binary classifiers makes them appealing, and thus many methods have been proposed in the literature to rectify imbalance in order to utilize them. Chapter 3 explores the effect of imbalance on the performance of binary classifiers. A key insight gained is that high levels of imbalance affect the performance of binary classifiers severely, and even sampling methods are unable to boost the performance. One-class classifiers, in many cases, outperform the best performing binary classifier boosted by sampling. However, if the domain under consideration is complex, then this edge disappears. This complexity, we argue, results from the presence of sub-concepts that are being ignored, and in Chapter 4, we develop a framework that exploits knowledge of these sub-concepts to induce effective one-class classifiers.

8.1 Simplification with domain knowledge

More often than not we have at least a modicum of domain specific knowledge to work with. In such cases, we identify, with the assistance of a domain expert, the sub-concepts that can best partition the domain.

Our framework is perhaps best illustrated in the domain of behavioural biometrics for user authentication on mobile devices, where we employ a users swipe as their biometric signature. This domain possess all the properties to make the development of a machine learning system a most challenging task. From a usability perspective, the authentication must commence with minimal training data, and provide high levels of accuracy. From a learning perspective, the data is massively imbalanced (as the only available data is from the user) comprised of multiple concepts, and susceptible to concept shifts and oscillations. In order to handle imbalance, we develop a simple yet computationally efficient one-class classifier. Concept shifts are handled by conducting online learning using a rolling window, rather than traditional batch learning. With respect to the concepts, we identify a users motion as the discriminator between them. For instance, swipes done while walking will be very different from swipes done while sitting. Modern smart phones come with a multitude of devices that enable us to ascertain the exact motion the user is engaged in. This facilitates labelling data without an external "teacher".

Yet another domain that we consider is the detection of invasive isotopes as represented by gamma-ray spectra, where we divide the data based on whether a spectrum was influenced due to the weather or not; visual inspection of randomly selected spectra measured during rainy and dry days showed an obvious distinction in shape. Division conducted in this manner naturally leads to clusters that are explicit on the concept they are meant to represent; clusters discovered blindly will tend to have some degree of overlap between concepts. Unlike the previous domain, where it is possible to explicitly identify the sub-concept, the hardware employed by Health Canada cannot detect whether a spectrum is being affected by weather. Thus, we train a binary classifier that discriminates over the sub-concepts. Specifically, we include them in

the division layer; given that we have explicit knowledge about the concepts within the domain, we can induce classifiers over these concepts. Thus, these classifiers take on the role of a clustering agent, as novel instances are processed by them and sent to the appropriate one-class classifiers. We employ the Naïve Bayes classifier in this layer. As expected, the resulting system outperformed the standard systems by a wide margin.

8.2 Simplification without domain knowledge

In the worst case scenario, we may not have sufficient knowledge to identify underlying sub-concepts in the domain. This is exemplified in the domain considered in our research project with Health Canada that involved attempting to induce classifiers that could detect clandestine nuclear tests being conducted by nations, as measured by concentrations of Xenon isotopes in the atmosphere. It is straightforward to understand why this data set would be imbalanced; obtaining samples relating to nuclear explosions is a very tricky proposition, as such tests do not typically occur often. This naturally made the domain an ideal candidate for one-class classification. Initial experiments with them, however, showed that attempting to model the entire data set led to poor classification results. We hypothesized that this was due to the data space being highly complex and multi-modal. Unfortunately, we did not have access to any domain knowledge with which we could identify the source of these complexities.

We rectified this issue by employing a basic clustering algorithm over a complex domain, and inducing models over each cluster. We observed that even this can lead to an improvement in performance when compared to a model induced over the domain as a whole. Clustering, by definition, divides the data space into regions where data instances in each cluster are more similar to each other than instances in other clusters. Each cluster, as a result, can be thought to represent a distinct, domain independent concept; the original space that represents a complex concept has been reduced to a number of simpler concepts. This naïve *divide-and-conquer* approach

results in an ensemble of simpler models that collectively outperform a single, more complex model, as was evident in the performance of our system over the CTBT data set.

8.3 Revisiting the thesis contribution

The thesis argues that certain domains exhibit such extreme levels of imbalance that one-class classification is the only learning paradigm that can be effectively employed (as discussed in Chapter 3). However, as we note in Chapter 4, overlap between classes and multi-modality in the domain can be prohibitive to the performance of one-class classifiers. We hypothesize that these complexities arise to the presence of implicit concepts (sub-concepts) within the domain, and the performance of one-class classifiers can be vastly improved by employing domain specific knowledge to identify and learn over these implicit concepts. In particular, we propose two approaches:

1. The case when there is domain knowledge: Knowledge of the intricacies of the domain implies that we have a notion on the nature of the underlying sub-concepts. In this case we propose to explicitly divide the data based on the these sub-concepts. Two considerations need to be taken into account in this case:

    - Identification of sub-concepts is directly possible. In other words, it is is possible to identify the sub-concept to which a novel instance belongs with guaranteed 100% accuracy. This is evident in the biometric domain considered, as a mobile device's hardware is capable of identifying a users state of motion.
    - Identification of sub-concepts is not directly possible. This case, we propose inducing a multi-class classifier over the inherent sub-concepts; novel samples are passed to the appropriate one-class classifier as classified by the multi-class classifier. Instead of classifying using all the classifiers in the ensemble, only the classifier over the concept selected by the multi-class

classifier is used.

2. The case of no domain knowledge: In this case, we propose to simply cluster the domain, and build one-class classifiers over these clusters. The resulting ensemble makes the classification decision collectively.

Our work was motivated by, and tested on, real-world problems. The domains we utilized exemplified the variety and quality of data that is encountered in practice. The results presented in this thesis offer strong support for our framework. In particular, the most significant insight to be gained is that, if possible, it is useful to exploit the nuances of the domain for producing efficient classification systems.

8.4 Future Work

Our work introduces a framework; as with any framework, there will be scope for improvement within its various components. In this section we discuss promising avenues for further work.

8.4.1 Feature Selection for sub-concepts

In the experiments conducted in this thesis, we utilize the same set of features for all sub-concept. However, in certain domains, it could well be the case that a unique set of features would be associated with each sub-concept, and inducing models by utilizing only these features would be more effective than utilizing all features. In other words, subsequent to identifying sub-concepts, prior to training one-class classifiers, we would identify the most appropriate features for the sub-concepts, and then employ the appropriate learning method as presented in this thesis. Thus, exploring feature selection for sub-concepts presents an important avenue for future work.

8.4.2 Combining frameworks

While we present three different frameworks, it is possible to combine them, depending on the application. For instance, in the complete knowledge approach, we know *a*

priori what sub-concept the datum belongs too. In the fuzzy knowledge approach we utlize a multi-class classifier to learn how to discriminate between concepts. We can produce a new framework in which we merge these two approaches together; we can use the power that discrimination algorithms offer to boost the complete knowledge approach. Specifically, we can induce a probabilistic classifier to assign a classification to the novel datum. Given that we have knowledge as to the sub-concept to which it belongs, we can check if the classification probability assigned by the classifier to the datum for that sub-concept is above or below a pre-defined threshold. If it is below, we can classify that as being an outlier, rather than further passing it along for processing by one-class classifiers.

This is but one approach for combining the two frameworks. Depending on the nuances of the domain, it it possible to come up with different combinations; the onus of how and what to combine will depend on the experience and knowledge of both the domain expert and the machine learning researcher.

8.4.3 Classifier ensembles in the absence of knowledge

With respect to the the framework presented in the absence of domain knowledge, there are several interesting directions for future research into the use of clustering for one-class classification. For the experiments presented in this thesis, we used two clustering algorithms: the k-means algorithm and the k-medoids algorithm. The success of these algorithms is highly dependent on selecting an appropriate value of k. As this information is not always easily available, depending on the domain, it would be worth exploring other clustering algorithms, such as the expectation-maximization (EM) algorithm, density based algorithm such as DBScan, or even hierarchical clustering, for discovering clusters. The dendrogram from the hierarchical clustering could give useful insight into the presence of sub-concepts and be used to determine the appropriate of clusters to employ. It is also likely that, depending on the problem domain, an ensemble of one-class classifiers can be used on the clusters, where a different classifier is trained on each cluster.

8.4.4 One-class classifiers as substitutes for binary classifiers

Typically, one would employ one-class classifiers over a domain only if the imbalance between the two classes is prohibitive for the application of binary classifiers. However, during our work we noted that the one-class classifiers, when trained under our frameworks, outperformed the binary classifiers complimented by sampling even at moderate levels of imbalance in the sonar, alphabets and all three forest cover domains. This is an interesting and exciting observation as it opens the door for further exploration into the application of one-class classifiers even when the levels of imbalance are not extreme. Depending on the properties of the domain, one could possibly achieve better classification results by employing a one-class classifier trained under the appropriate framework than a binary classifier.

8.4.5 Learning with multiple classes with Imbalance

The work in this thesis looks at binary and one-class classifiers. One-class classifiers fit naturally as a substitute for learning problems that come under a binary formulation when data exists only for one class; one of the two classes becomes the target and the other the outlier. Our work, then, looks at a specific case of multi-class learning. However, things are not as straightforward when there the domain has multiple classes. Consider a domain with five classes, and data is only available from three classes. Given the domain formulation, it may not be prudent to consider the two unknown classes as belonging to a single outlier class, as knowing to know which of the 5 classes a novel datum belongs to could be important. Applying a one-class classifier directly would be not appropriate, as it would treat the two unknown classes as a single class. Furthermore, inducing a multi-class classifier over the known classes would also not be appropriate as it would simply learn to discriminate between the three classes, and not take into account that the domain has two other classes. One possibility to resolve the latter aspect would be to apply one of our frameworks over the known classes, treating each class as a sub-concept and inducing a one-class classifier over them. However, dealing with the unknown classes would be a challenge,

especially if the data is impossible to collect or reliably simulate. Thus, inducing efficient classification systems over multi-class domains that suffer from imbalance is an important and challenging avenue of research.

8.5 A Final Note

This thesis has introduced a framework for learning with one-class classifiers over domains that exhibit complexity resulting from multi-modality and overlap. We demonstrate that these complexities can be resolved by identifying distinct sub-concepts that underlie the domains, using them to simplify the space, and then modelling each sub-concept with a one-class classifier. Access to, and understanding of, domain knowledge is key, and to this end, we introduce three distinct frameworks, each dependent on how the domain knowledge can be utilized. We apply each of these frameworks over three real-world domains, and demonstrate that these frameworks are able to produce powerful classification systems. All the domains presented in the thesis conform to a two-class formulation. We hope that future research will attempt to expand our framework to deal with mutli-class domains, and explore the various avenues discussed in this chapter.

Appendix A

Full opacity plots for the Alphabets, Forest, ForestC1 and ForestC2C5 datasets

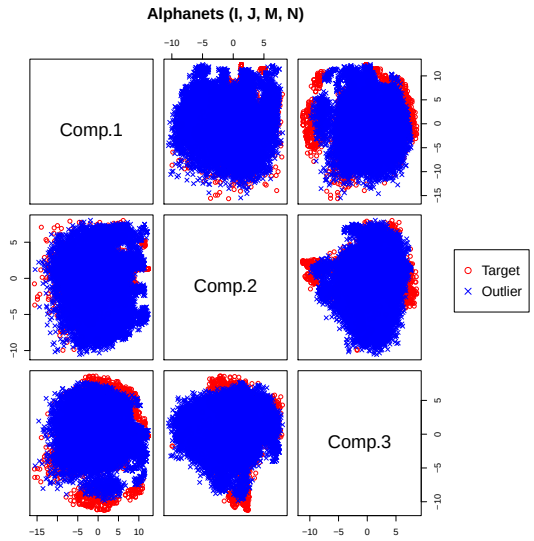


Figure A.1: The first three principle components of the *Alphabets* dataset.

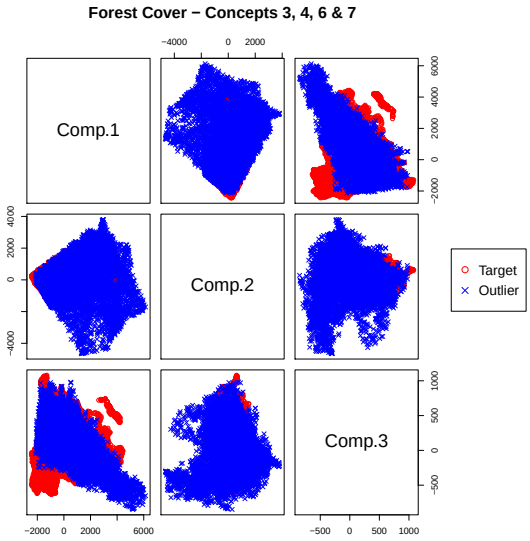


Figure A.2: The first three principle components of the *Forest* dataset.

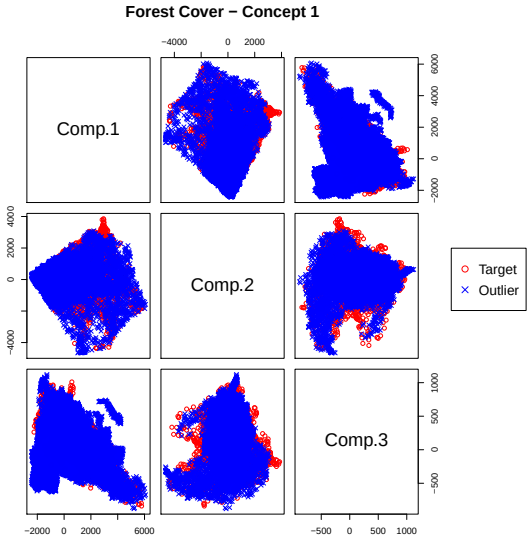


Figure A.3: The first three principle components of the *ForestC1* dataset.

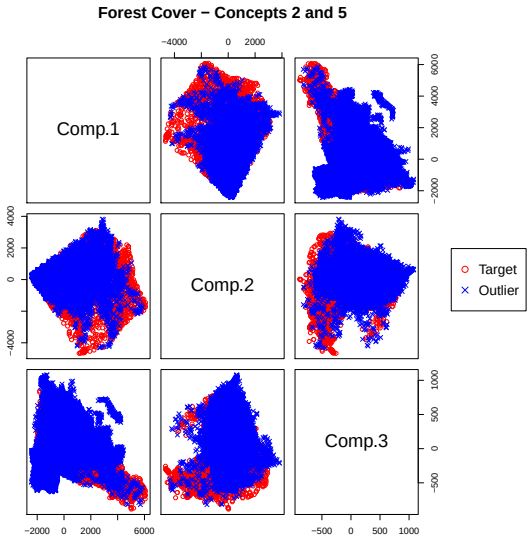


Figure A.4: The first three principle components of the *ForestC2C5* dataset.

Appendix B

Sampling plots for binary classifiers

Here we include the plots that display the trends in classifier performance for all the sampling methods not considered in Chapter 3.

B.1 Results on *Artificial Unimodal* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the artificial unimodal dataset.

B.2 Results on *Artificial Multimodal* dataset with no overlap

This section contains the plots for oversampling, one-sided selection with Tomek Links and undersampling for the artificial multimodal dataset.

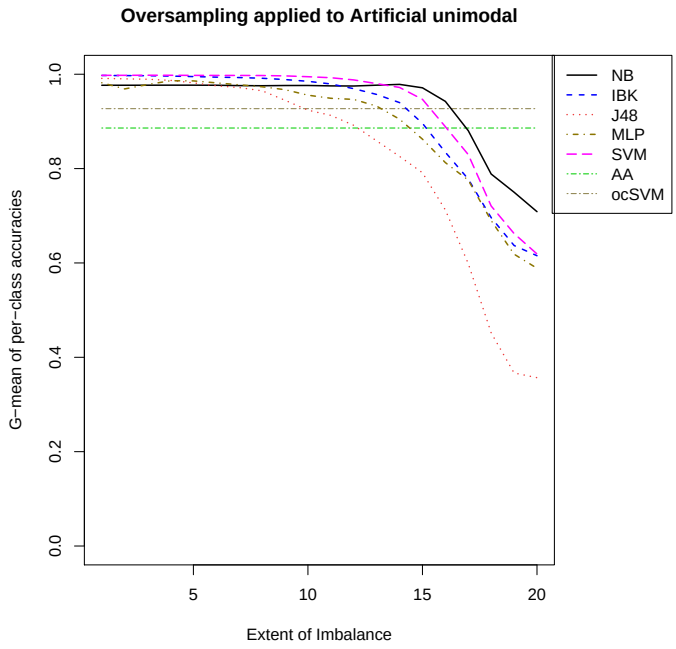


Figure B.1: The performance trends of various classifiers over the artificial unimodal dataset with oversampling.

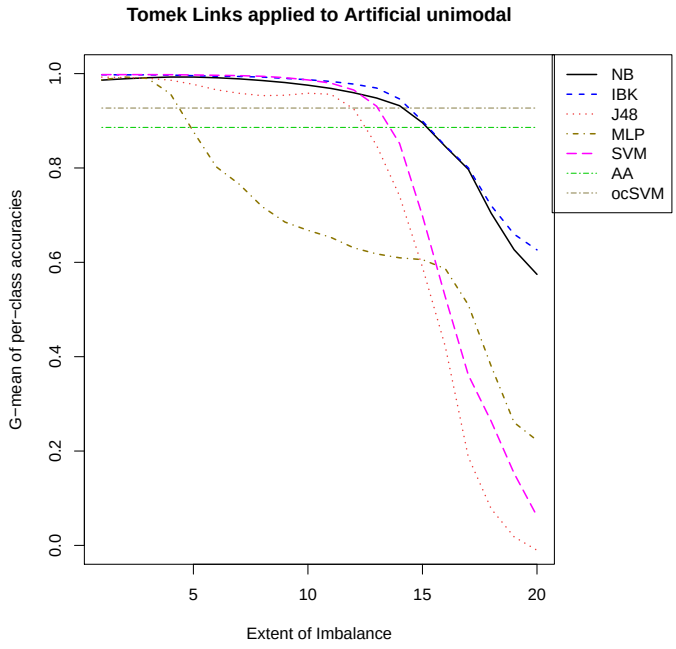


Figure B.2: The performance trends of various classifiers over the artificial unimodal dataset with one-sided selection.

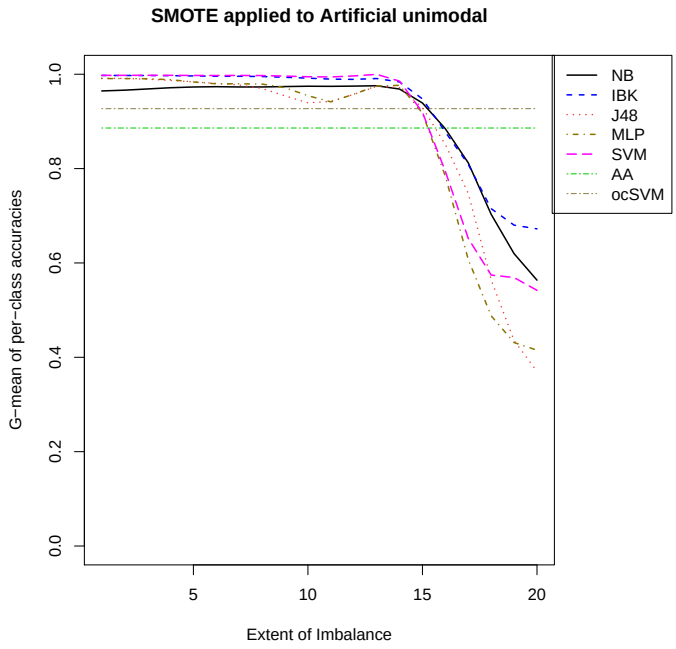


Figure B.3: The performance trends of various classifiers over the artificial unimodal dataset with SMOTE.

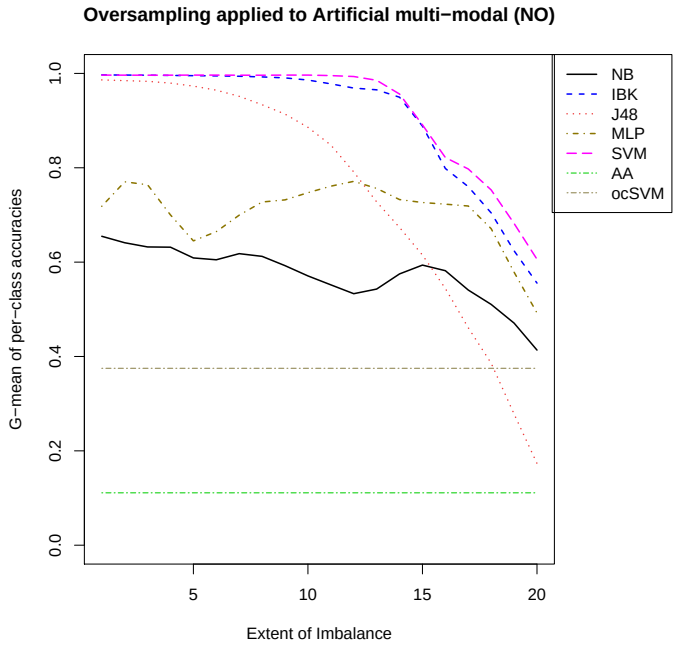


Figure B.4: The performance trends of various classifiers over the artificial multimodal dataset with oversampling.

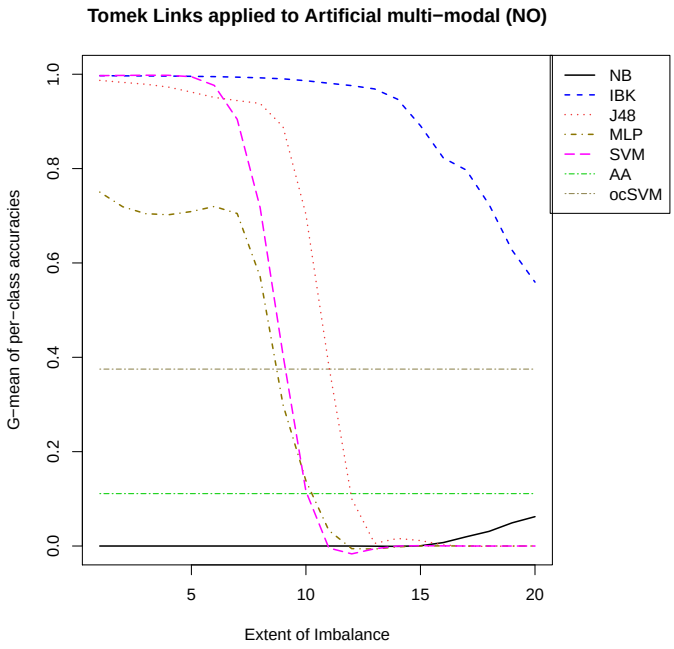


Figure B.5: The performance trends of various classifiers over the artificial multimodal dataset with one-sided selection.

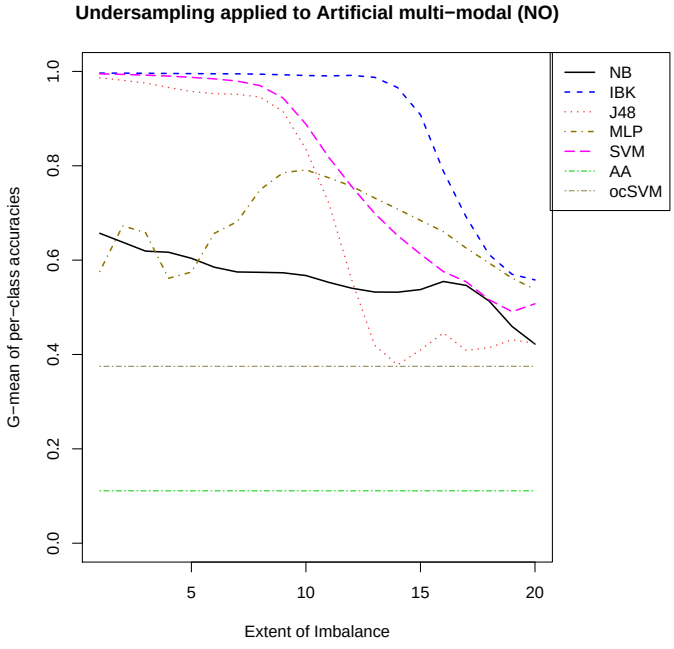


Figure B.6: The performance trends of various classifiers over the artificial multimodal dataset with undersampling.

B.3 Results on *Artificial Multimodal* dataset with overlap

This section contains the plots for oversampling, one-sided selection with Tomek Links and undersampling for the artificial multimodal dataset.

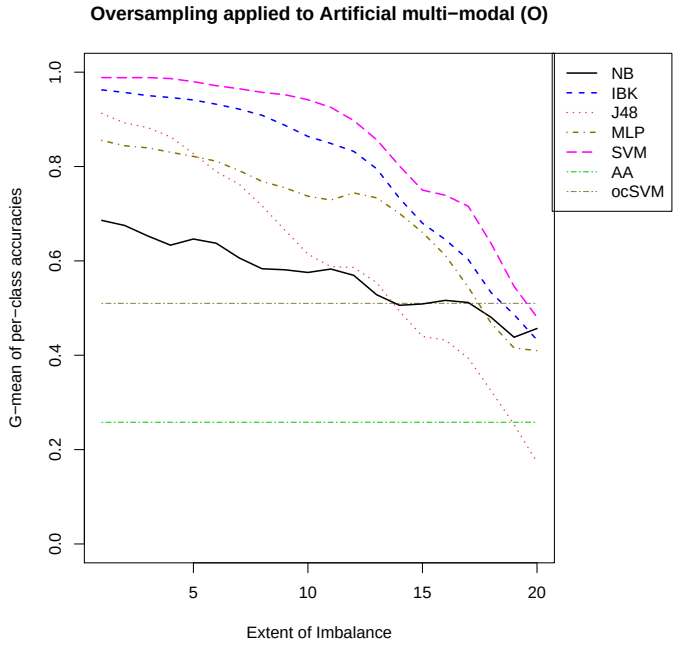


Figure B.7: The performance trends of various classifiers over the artificial multimodal dataset with oversampling.

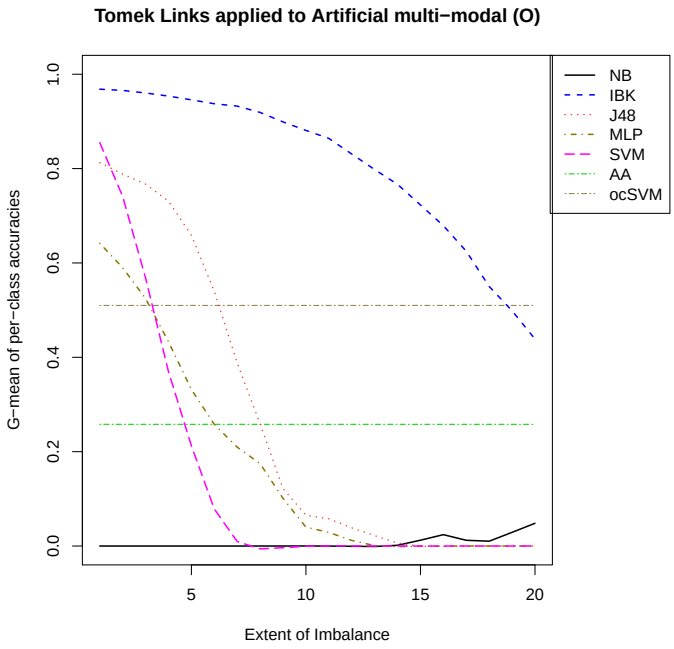


Figure B.8: The performance trends of various classifiers over the artificial multimodal dataset with one-sided selection.

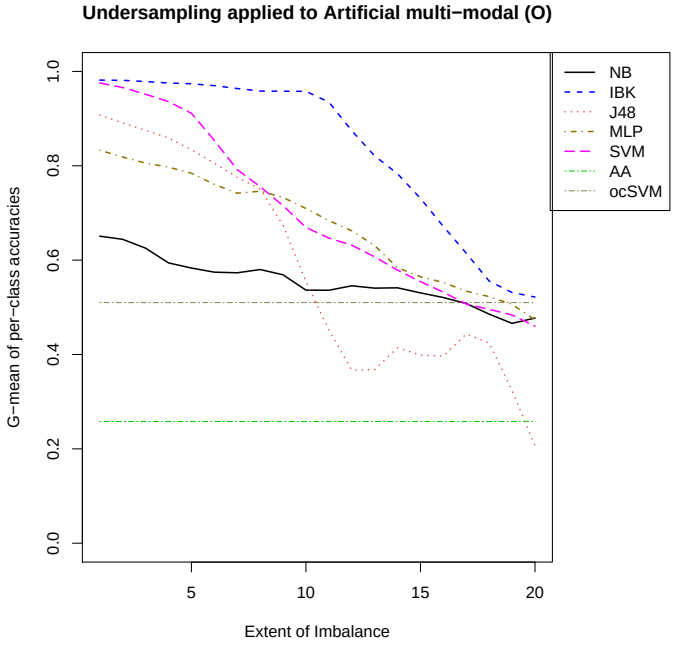


Figure B.9: The performance trends of various classifiers over the artificial multimodal dataset with undersampling.

B.4 Results on *Diabetes* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the diabetes dataset.

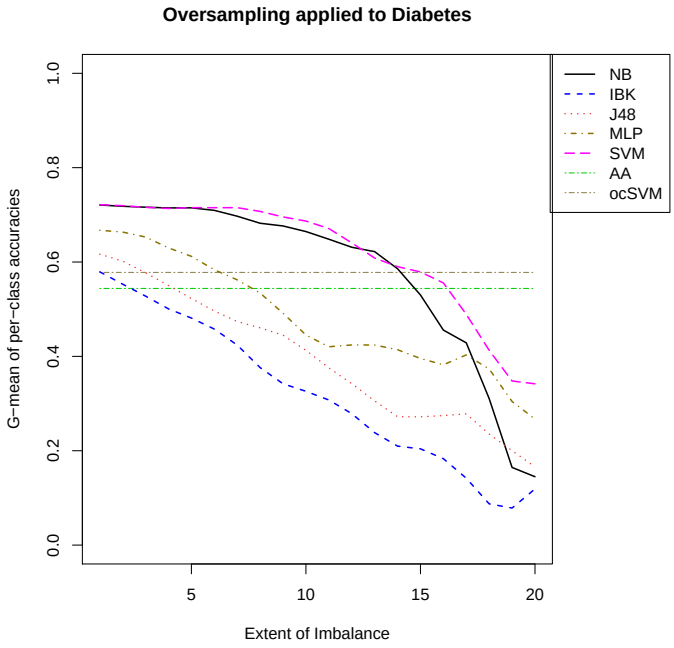


Figure B.10: The performance trends of various classifiers over the diabetes dataset with oversampling.

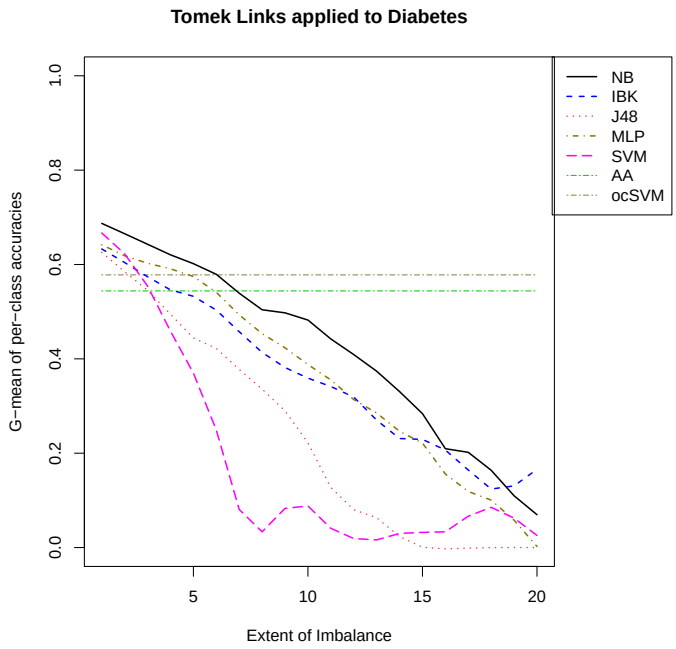


Figure B.11: The performance trends of various classifiers over the diabetes dataset with one-sided selection.

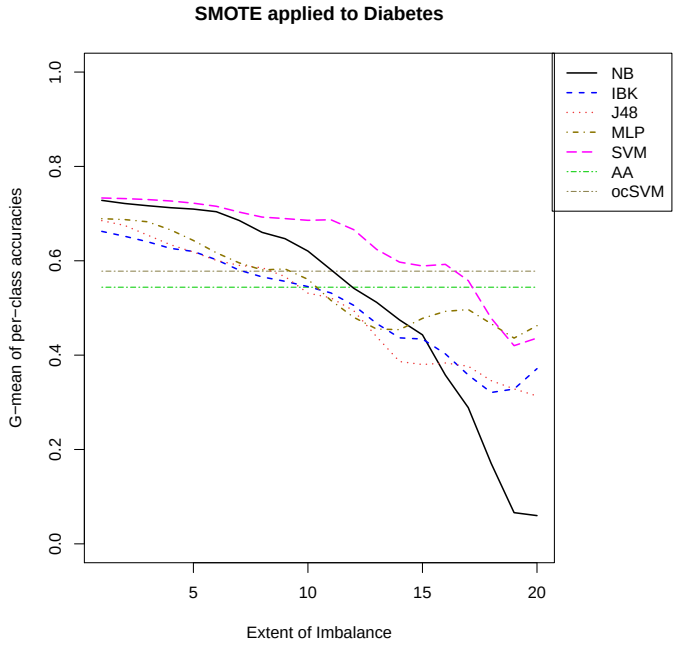


Figure B.12: The performance trends of various classifiers over the diabetes dataset with SMOTE.

B.5 Results on *Heart Disease* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the heart disease dataset.

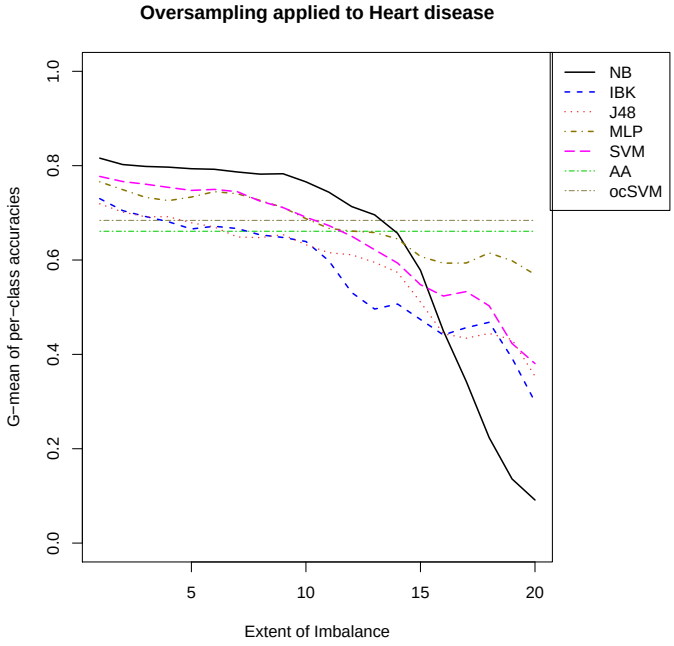


Figure B.13: The performance trends of various classifiers over the heart disease dataset with oversampling.

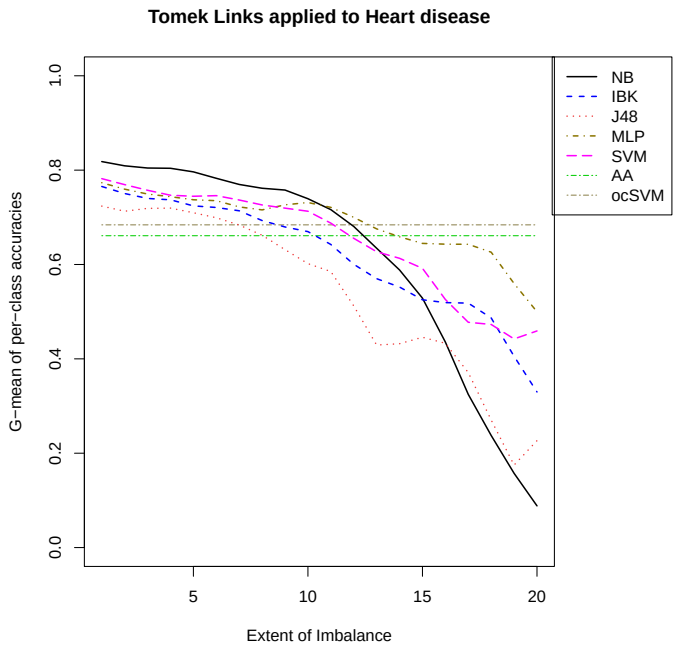


Figure B.14: The performance trends of various classifiers over the heart disease dataset with one-sided selection.

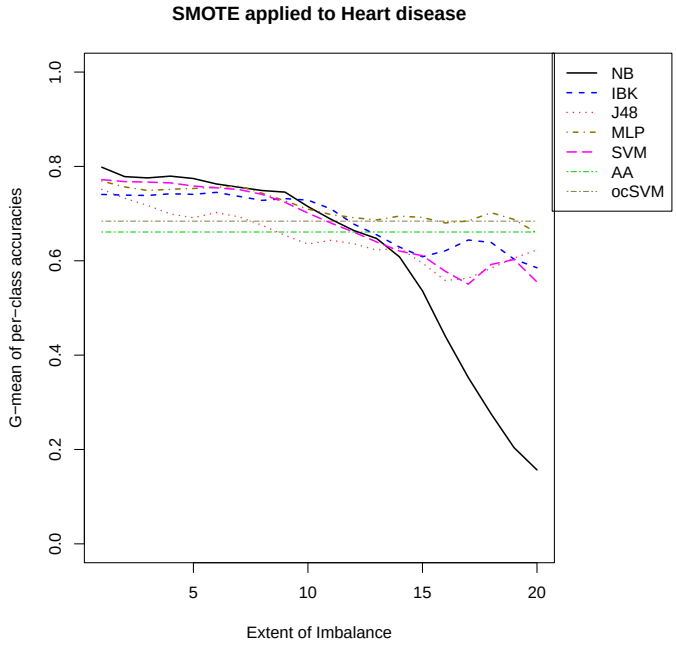


Figure B.15: The performance trends of various classifiers over the heart disease dataset with SMOTE.

B.6 Results on *Ionosphere* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the ionosphere dataset.

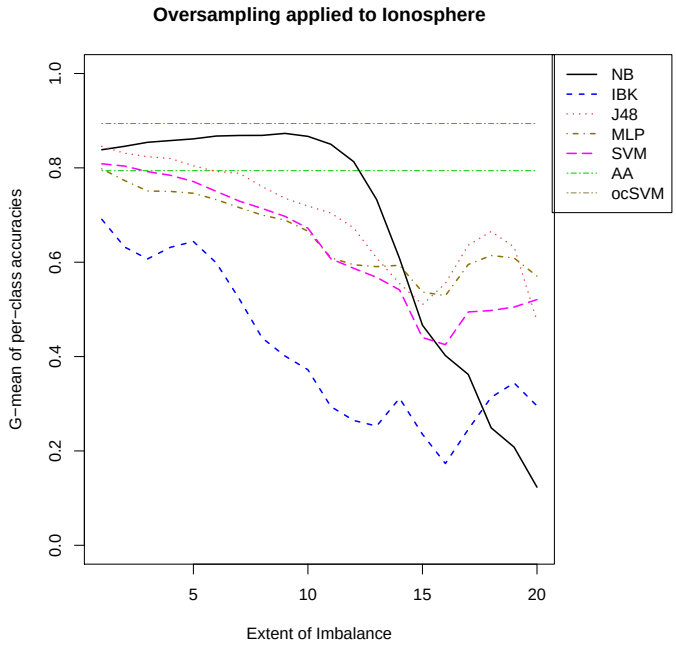


Figure B.16: The performance trends of various classifiers over the ionosphere dataset with oversampling.

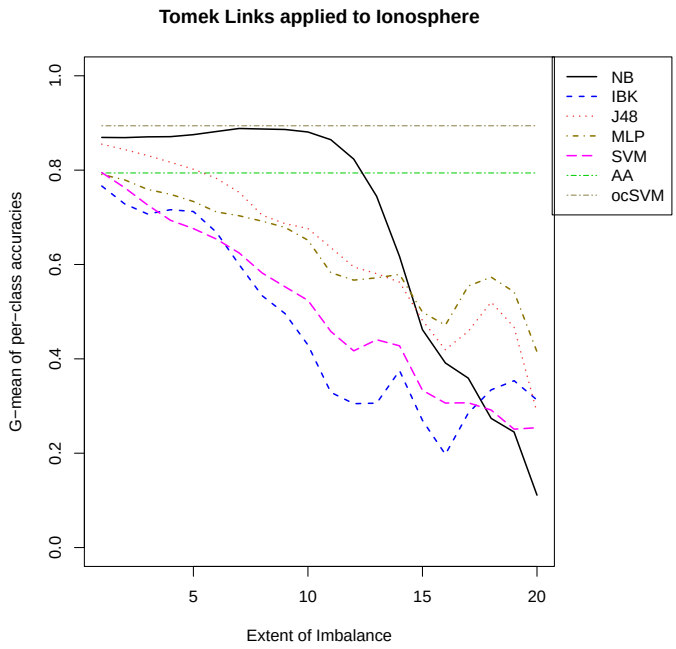


Figure B.17: The performance trends of various classifiers over the ionosphere dataset with one-sided selection.

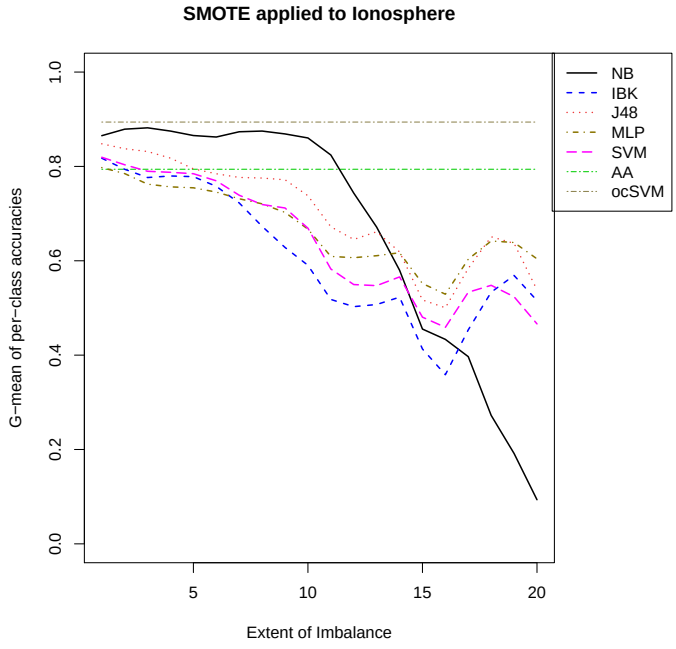


Figure B.18: The performance trends of various classifiers over the ionosphere dataset with SMOTE.

B.7 Results on *Sonar* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the sonar dataset.

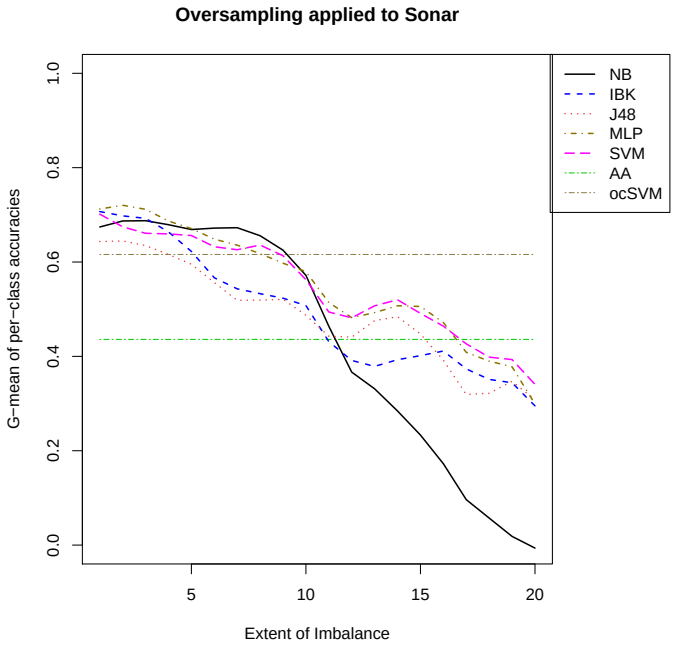


Figure B.19: The performance trends of various classifiers over the sonar dataset with oversampling.

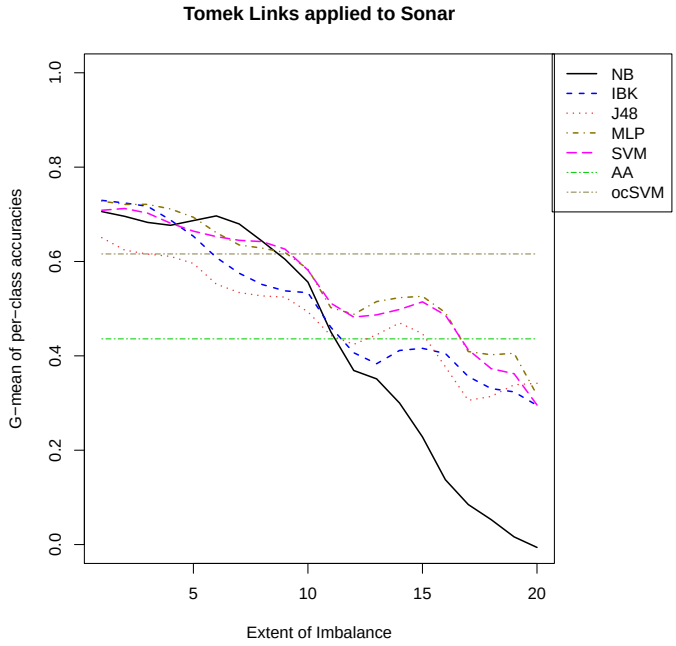


Figure B.20: The performance trends of various classifiers over the sonar dataset with one-sided selection.

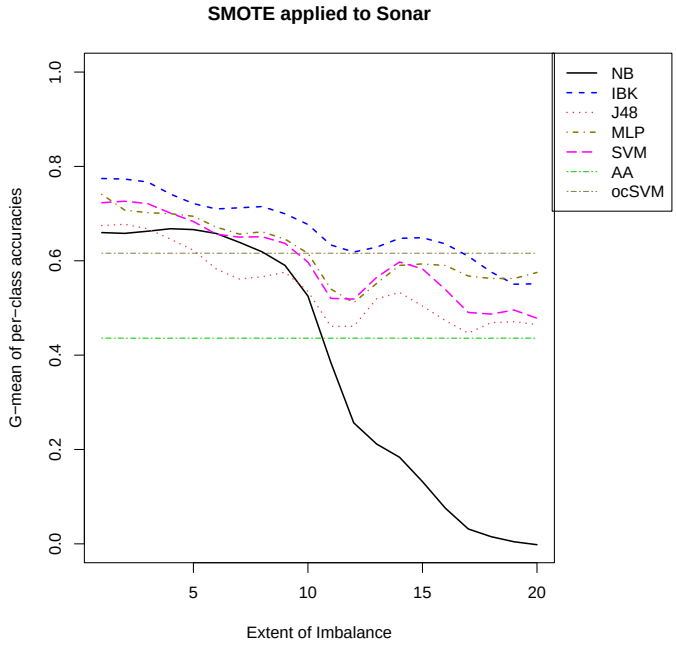


Figure B.21: The performance trends of various classifiers over the sonar dataset with SMOTE.

B.8 Results on *Thyroid Disease* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the thyroid disease dataset.

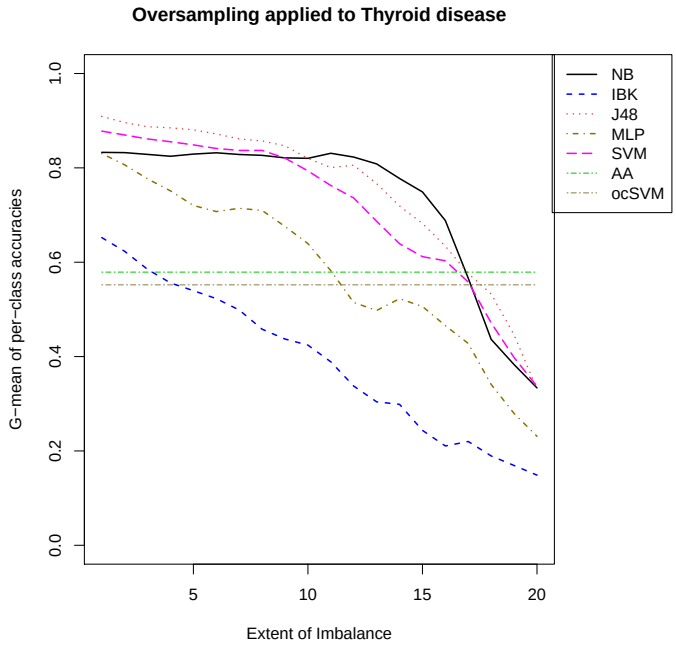


Figure B.22: The performance trends of various classifiers over the thyroid disease dataset with oversampling.

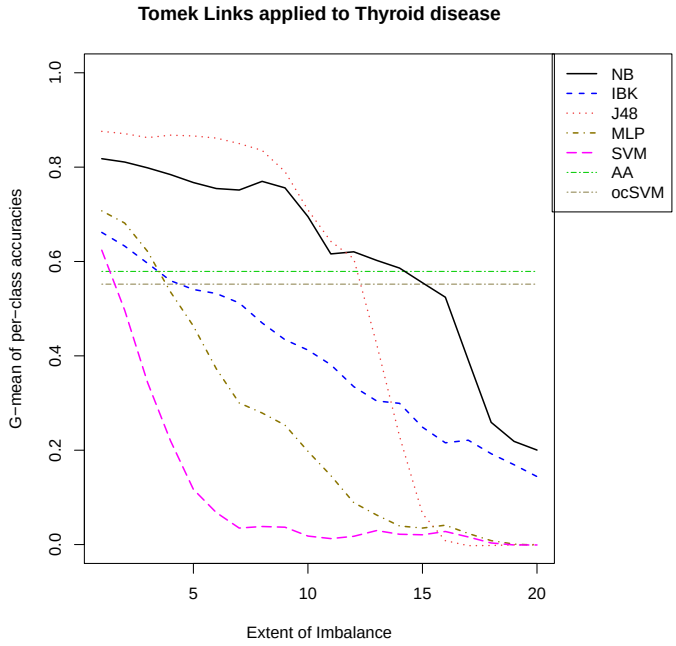


Figure B.23: The performance trends of various classifiers over the thyroid disease dataset with one-sided selection.

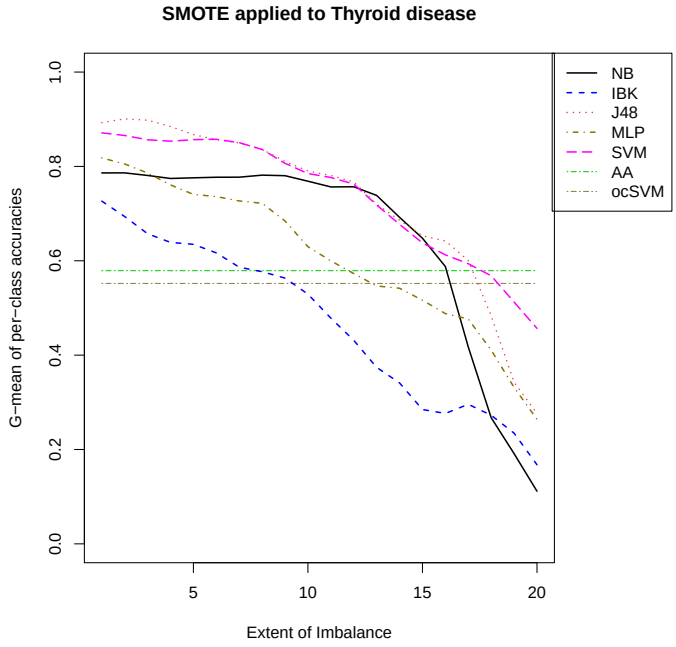


Figure B.24: The performance trends of various classifiers over the thyroid disease dataset with SMOTE.

B.9 Results on *Alphabets* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the alphabets dataset.

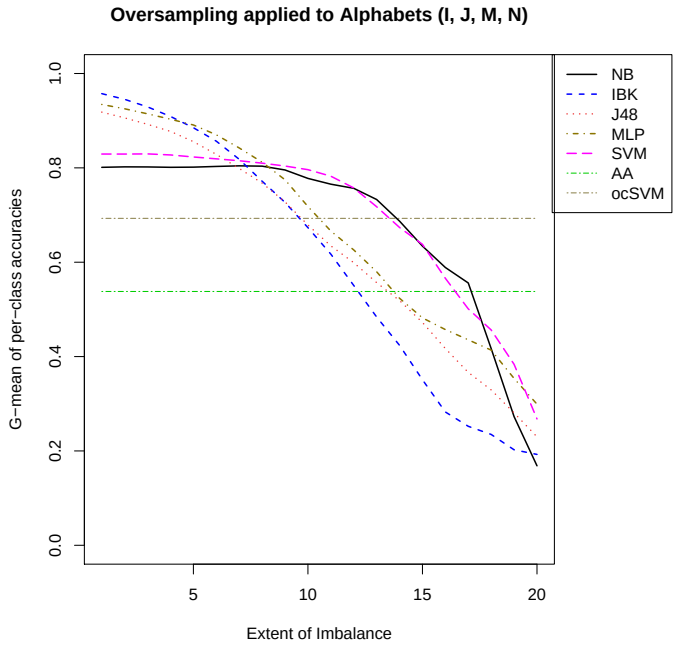


Figure B.25: The performance trends of various classifiers over the alphabets dataset with oversampling.

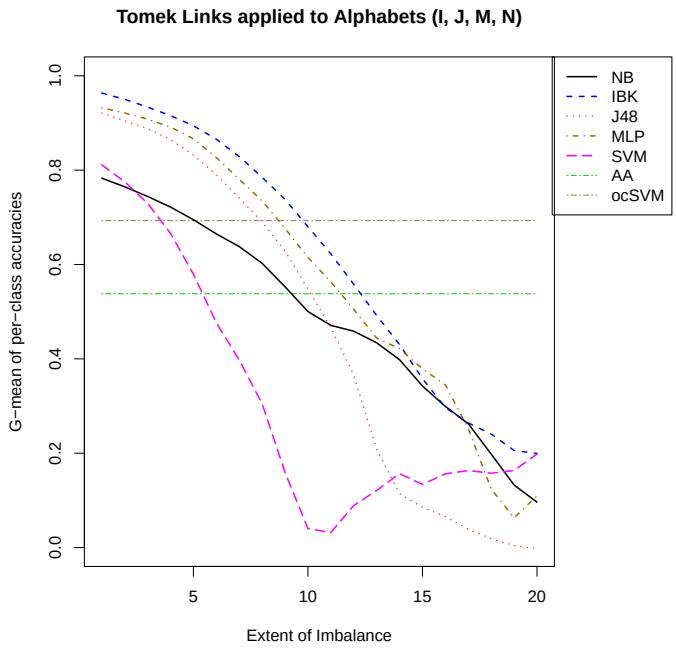


Figure B.26: The performance trends of various classifiers over the alphabets dataset with one-sided selection.

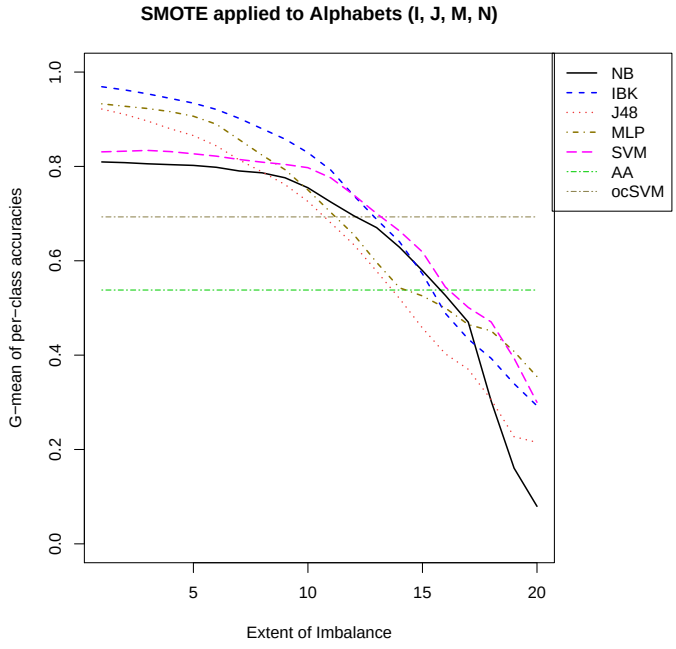


Figure B.27: The performance trends of various classifiers over the alphabets dataset with SMOTE.

B.10 Results on *Forest* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the forest dataset.

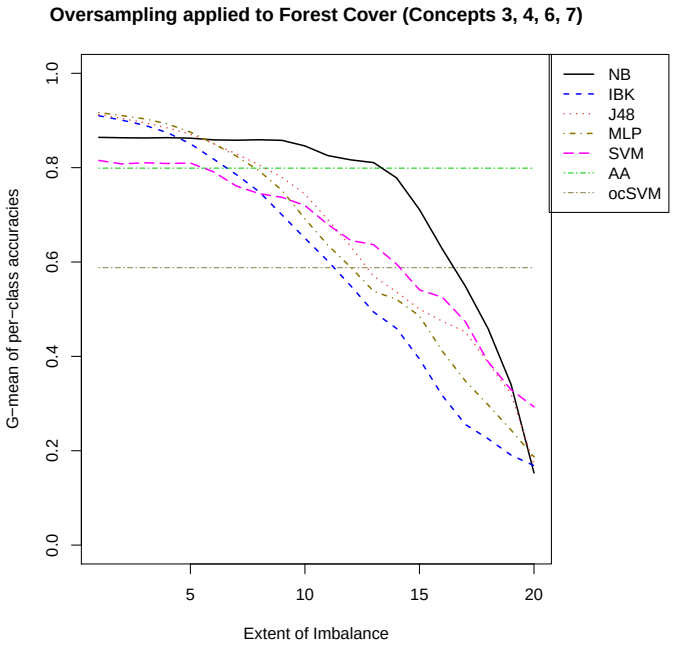


Figure B.28: The performance trends of various classifiers over the forest dataset with oversampling.

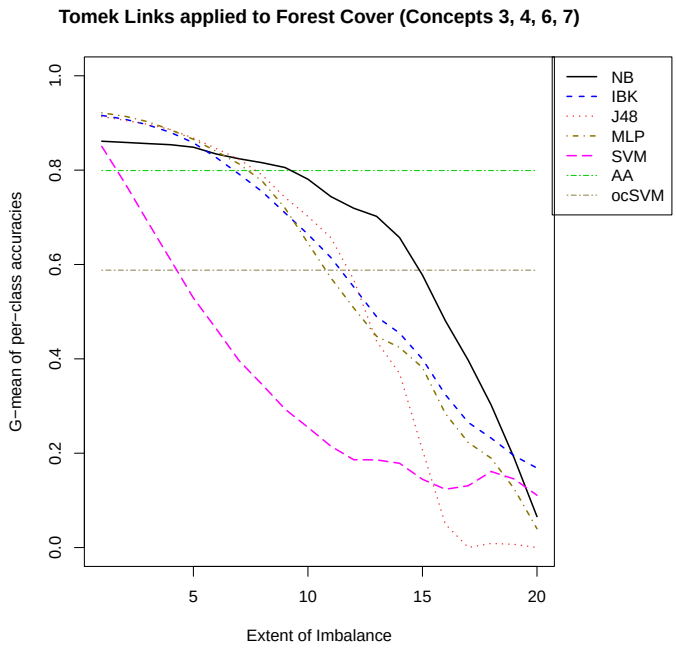


Figure B.29: The performance trends of various classifiers over the forest dataset with one-sided selection.

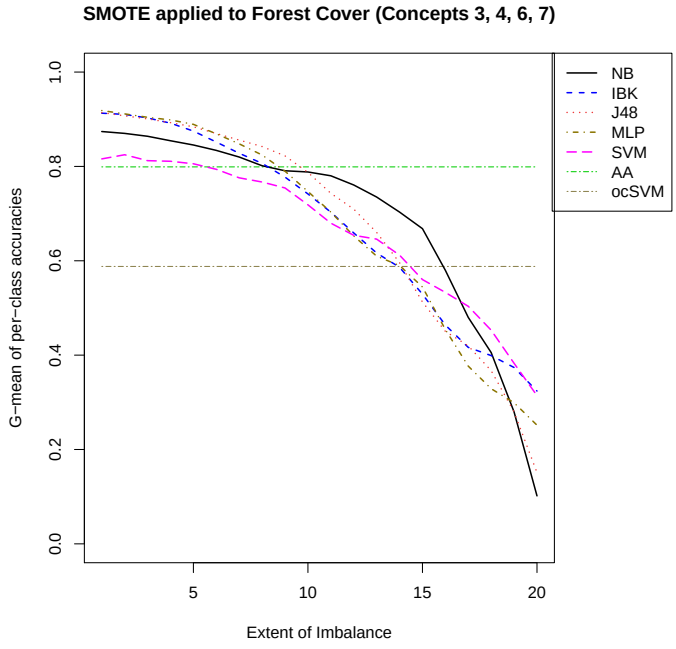


Figure B.30: The performance trends of various classifiers over the forest dataset with SMOTE.

B.11 Results on *ForestC1* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the forestC1 dataset.

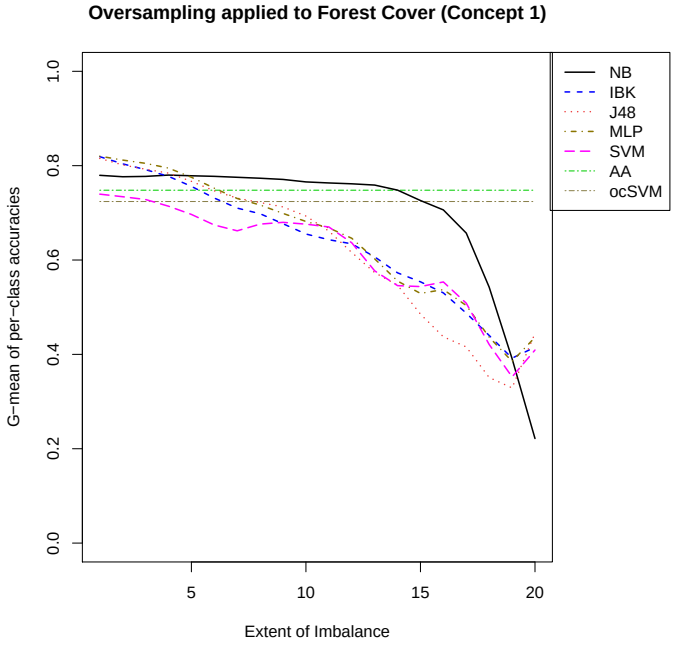


Figure B.31: The performance trends of various classifiers over the forestC1 dataset with oversampling.

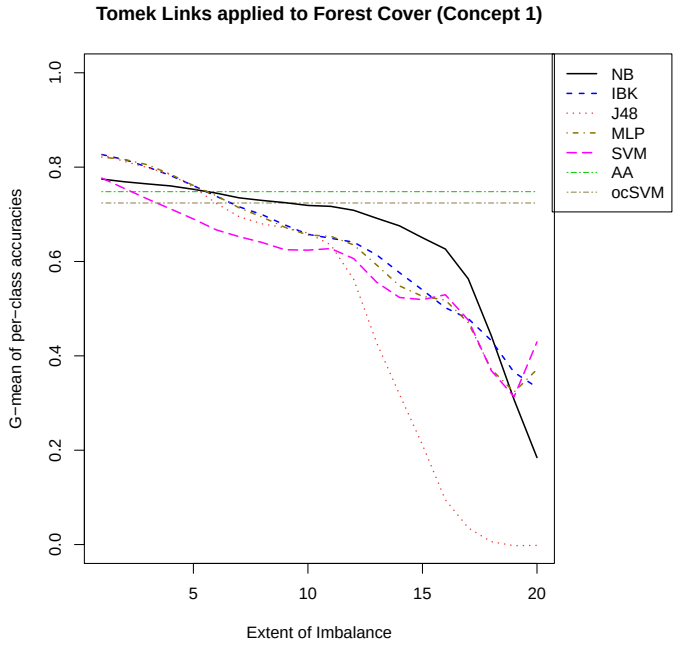


Figure B.32: The performance trends of various classifiers over the forestC1 dataset with one-sided selection.

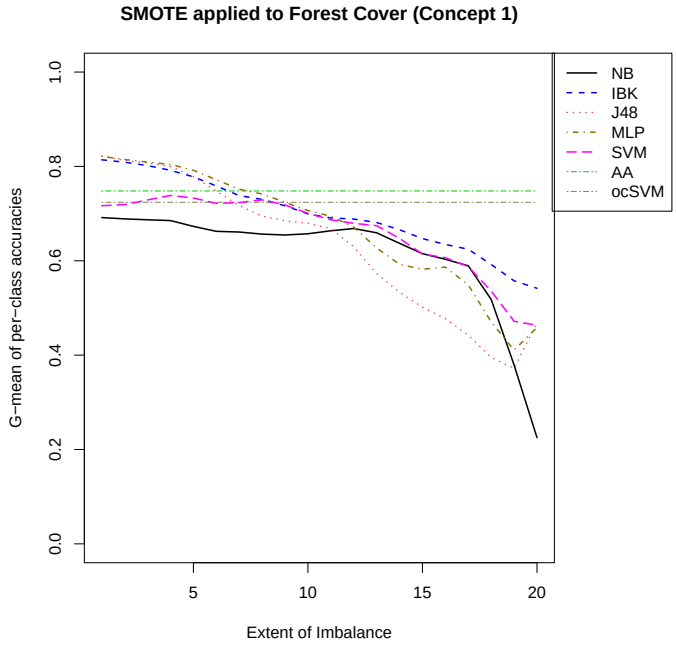


Figure B.33: The performance trends of various classifiers over the forestC1 dataset with SMOTE.

B.12 Results on *ForestC2C5* dataset

This section contains the plots for oversampling, one-sided selection with Tomek Links and SMOTE for the forestC2C5 dataset.

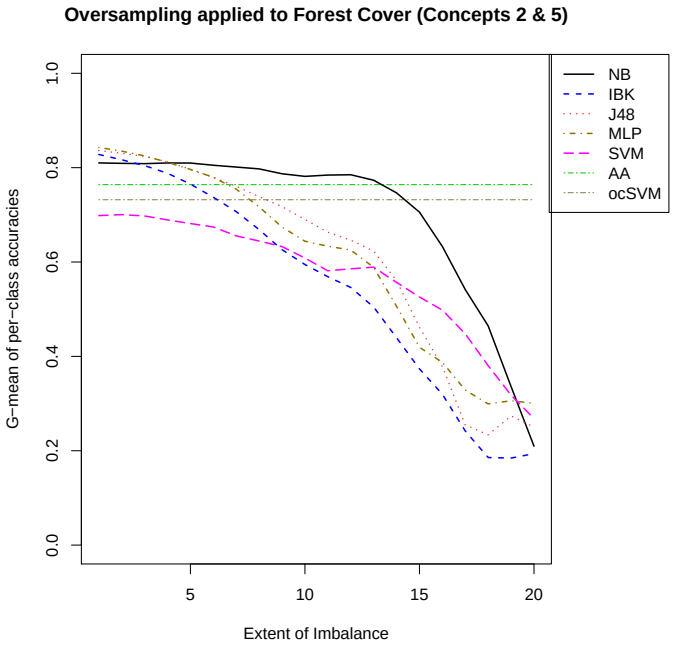


Figure B.34: The performance trends of various classifiers over the forestC2C5 dataset with oversampling.

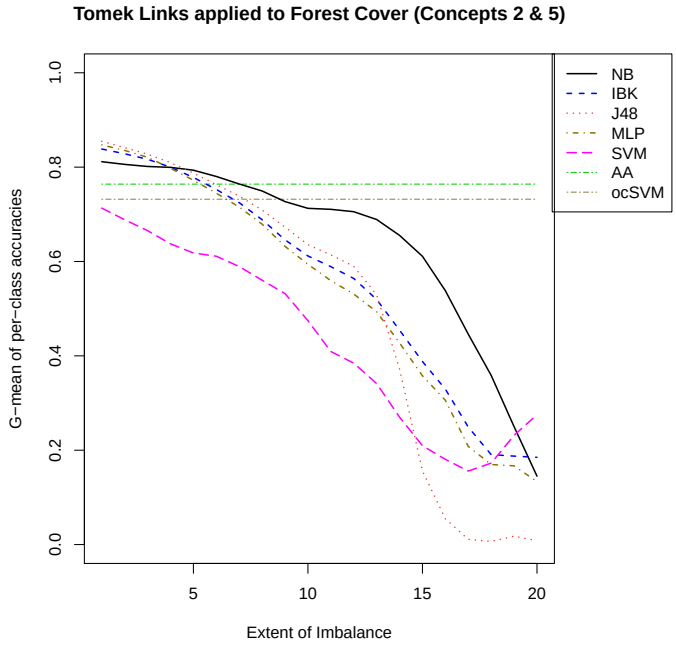


Figure B.35: The performance trends of various classifiers over the forestC2C5 dataset with one-sided selection.

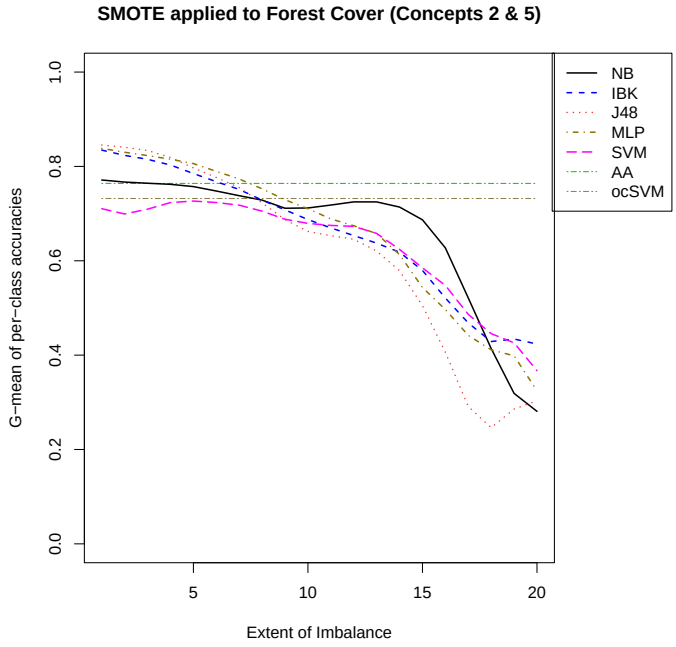


Figure B.36: The performance trends of various classifiers over the forestC2C5 dataset with SMOTE.

Bibliography

[1] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. In *In Proceedings of the 15th European Conference on Machine Learning (ECML*, pages 39–50, 2004.

[2] S. P. Arya. *Air Pollution Meteorology and Dispersion.* Oxford University Press New York, NY, 1999.

[3] S. Awasthi, M. Khare, and P. Gargava. General Plume Dispersion Model (GPDM) for point source emission. *Environmental Modeling and Assessment*, 11(3):267–276, 2006.

[4] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1):20–29, 2004.

[5] C Bellinger and N Japkowicz. Motivating the inclusion of meteorological indicators in the ctbt feature-space. In *Proceedings of IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2011.

[6] C. Bellinger and B. J. Oommen. On simulating episodic events against a background of noise-like non-episodic events. In *42nd Summer Computer Simulation Conference, SCSC 2010, Ottawa, Canada, July 11-14, 2010. Proceedings*, 2010.

[7] C. Bellinger and B. J. Oommen. On the pattern recognition and classification of stochastically episodic events. *Transactions on Computational Collective Intelligence*, Accepted for Publication, 2011.

[8] Colin Bellinger, Shantanu Sharma, and Nathalie Japkowicz. One-class versus binary classification: Which and when? In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, pages 102–106, 2012.

[9] Brain Leke Betechuoh, Tshilidzi Marwala, and Thando Tettey. Autoencoder networks for hiv classification. *CURRENT SCIENCE-BANGALORE-*, 91(11):1467, 2006.

[10] Michael Bingham. Towards effective behavioural biometrics for mobile devices, 2015.

[11] J. F. Bowers, J. R. Bjorklund, and C. S. Cheney. Industrial Source Complex (ISC) dispersion model user's guide. Technical Report EPA-450/4-79-030, US Environmental Protection Agency, Research Triangle Park, NC, 1979.

[12] V Bulitko, R Greiner, R Kube, and W Zhou. Using autoencoding networks for tramp metal detection. *University of Alberta July*, 31, 2000.

[13] Philip K. Chan, Wei Fan, Andreas Prodromidis, and Salvatore J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, 14:67–74, 1999.

[14] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[15] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *Knowledge Discovery in Databases: PKDD 2003*, pages 107–119. Springer, 2003.

[16] Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. One-class svm for learning in image retrieval. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 34–37. IEEE, 2001.

[17] Jun Deng, Zixing Zhang, Erik Marchi, and Bjorn Schuller. Sparse autoencoder-based feature transfer learning for speech emotion recognition. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 511–516. IEEE, 2013.

[18] Misha Denil and Thomas Trappenberg. Overlap versus imbalance. In *Advances in Artificial Intelligence*, pages 220–231. Springer, 2010.

[19] Chesner Désir, Simon Bernard, Caroline Petitjean, and Laurent Heutte. One class random forests. *Pattern Recognition*, 46(12):3490–3506, 2013.

[20] Richard O Duda and Peter E Hart. Pattern recognition and scene analysis, 1973.

[21] J.P. Fontaine, F. Pointurier, X. Blanchard, and T. Taffary. Atmospheric xenon radioactive isotope monitoring. *Journal of Environmental Radioactivity*, 72:129Ű–135, 2004.

[22] M. Frank, R. Biedert, and E. Ma. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *... Forensics and Security ...*, pages 1–1, July 2013.

[23] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. *ICML*, pages 148–156, 1996.

[24] Y. Freund and R.E. Schapire. A decision-theoretic generalization on on-line learning and an application to boosting. *Journal of Computer and System Sciences*, pages 119–139, 1997.

[25] Vicente García, Ramón Alberto Mollineda, and José Salvador Sánchez. On the k-nn performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications*, 11(3-4):269–280, 2008.

[26] Giorgio Giacinto, Roberto Perdisci, Mauro Del Rio, and Fabio Roli. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion*, 9(1):69–82, 2008.

[27] Giorgio Giacinto, Fabio Roli, and Luca Didaci. A modular multiple classifier system for the detection of intrusions in computer networks. In *Multiple Classifier Systems*, pages 346–355. Springer, 2003.

[28] Vladimir Golovko and Leanid Vaitsekhovich. Neural network techniques for intrusion detection. In *Proc. Int. Conf. Neural Networks and Artificial Intelligence*, pages 65–69, 2006.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[30] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *Advances in intelligent computing*, pages 878–887. Springer, 2005.

[31] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009.

[32] Kathryn Hempstalk. *Continuous Typist Verification using Machine Learning*. PhD thesis, Department of Computer Science, University of Waikato, 2009.

[33] Kathryn Hempstalk, Eibe Frank, and Ian H. Witten. One-class classification by combining density and class probability estimation. In *Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *Lecture Notes in Computer Science*, pages 505–519, Berlin, 2008. Springer.

[34] Kenneth L. Ingham and Anil Somayaji. A methodology for designing accurate anomaly detection systems, 2007.

[35] R. A. Jacobs, M.I. Jordan, S.J. Nolan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, pages 79–87, 1991.

[36] Markus Jakobsson. Implicit Authentication for Mobile Devices. 2009.

[37] N Japkowicz. *Concept-Learning in the Absence of Counter-Examples: An Autoassociation-Based Approach to Classication*. PhD thesis, Rutgers University, 1999.

[38] N Japkowicz. Class imbalances: Are we focusing on the right issue? pages 17–23, 2003.

[39] N Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6:429–450, 2002.

[40] Nathalie Japkowicz. Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning Volume 42, Issue 1/2*, 42:97–122, 2001.

[41] Nathalie Japkowicz. Supervised learning with unsupervised output separation. *IASTED International Conference on Artificial Intelligence and Soft Computing*, 2002.

[42] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *SIGKDD Explor. Newsl.*, 6(1):40–49, June 2004.

[43] Pilsung Kang and Sungzoon Cho. Eus svms: Ensemble of undersampled svms for data imbalance problems. In *In Lecture Notes in Computer Science*, 2006.

[44] E.M. Knorr and R. T. Ng. A unified approach for mining outliers. *Proc. Conf. of the Centre for Advanced Studies on Collaborative Research, Toronto*, 1997.

[45] Sarah Martina Kolly, Roger Wattenhofer, and Samuel Welten. A Personal Touch - Recognizing Users Based on Touch Screen Behavior Categories and Subject Descriptors.

[46] Adam Kowalczyk and Bhavani Raskutti. One class svm for yeast regulation prediction. *ACM SIGKDD Explorations Newsletter*, 4(2):99–100, 2002.

[47] Bartosz Krawczyk, Michał Woźniak, and Bogusław Cyganek. Clustering-based ensembles for one-class classification. *Information Sciences*, 264:182–195, 2014.

[48] H. P. Kriegel, M. Schubert, and Z. Arthur. Angle-based outlier detection in high-dimensional data. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 444–452, New York, NY, USA, 2008. ACM.

[49] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Outlier detection techniques. In *Tutorial at the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Washington, DC*, 2010.

[50] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *In Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

[51] Miroslav Kubat, Robert C. Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Mach. Learn.*, 30(2-3):195–215, February 1998.

[52] M.Z. Kukar and I.Konokenko. Cost-sensitive learning with neural networks. *ECAI*, pages 445–449, 1998.

[53] Kingsly Leung and Christopher Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342. Australian Computer Society, Inc., 2005.

[54] M. Lichman. UCI machine learning repository, 2013.

[55] Nedim Lipka, Benno Stein, and Maik Anderka. Cluster-based one-class ensemble for classification problems in information retrieval. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1041–1042. ACM, 2012.

[56] Y.H. Liu and Y.T Chen. Face recognition using total margin based adaptive fuzzy support vector machines. In *IEEE Transactions on Neural Networks*, volume 18, pages 178–192, 2007.

[57] Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. Touch me once and I know it?s you ! Implicit Authentication based on Touch Screen Patterns. pages 987–996, 2012.

[58] L. M. Manevitz and M. Yousef. One-class svms for document classification. *The Journal of Machine Learning Research*, 2:139–154, 2002.

[59] Saeed Masoudina and Reza Ebrahimpour. Mixture of experts: A literature survey. *Artificial Intelligence Review*, 2012.

[60] S. Matwin, A. Kouznetsov, D. Inkpen, O. Frunza, and P. O'Blenis. A new algorithm for reducing the workload of experts in performing systematic reviews. *Journal of the American Medical Informatics Association*, 17:446–453, 2010.

[61] Yuxin Meng, Duncan S Wong, Roman Schlegel, and Lam-for Kwok. Touch Gestures Based Biometric Authentication Scheme for Touchscreen Mobile Phones. pages 331–350, 2013.

[62] K. M. Mok, A. I. Miranda, K. U. Leong, and C. Borrego. A Gaussian puff model with optimal interpolation for air pollution modelling assessment. *International Journal of Environment and Pollution*, 35:111–137(27), 5 November 2008.

[63] Roberto Perdisci, Guofei Gu, and Wenke Lee. Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 488–498. IEEE, 2006.

[64] Ronaldo C. Prati, Gustavo E. A. P. A. Batista, and Maria C. Monard. Class imbalances versus class overlapping: An analysis of a learning system behavior, 2004.

[65] R. Bharat Rao, Sriram Krishnan, and Radu Stefan Niculescu. Data mining for improved cardiac care. *SIGKDD Explor. Newsl.*, 8(1):3–10, June 2006.

[66] UCI Machine Learning Reposity. Covertype dataset. `https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/covtype.info`.

[67] Premkumar Saravanan, Samuel Clarke, Duen Horng, Polo Chau, and Hongyuan Zha. LatentGesture : Active User Authentication through Background Touch Analysis. 2014.

[68] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[69] Holger Schwent and Maurice Milgram. Transformation invariant autoassociation with application to handwritten character recognition. *NIPS*, 1995.

[70] Shiven Sharma, Colin Bellinger, and Nathalie Japkowicz. Clustering based one-class classification for verification of the ctbt. *2012 Canadian AI*, 2012.

[71] Shiven Sharma, Colin Bellinger, Nathalie Japkowicz, Rodney Berg, and Kurt Ungar. Anomaly detection in gamma ray spectra: A machine learning perspective. In *Computational Intelligence for Security and Defence Applications (CISDA), 2012 IEEE Symposium on*, pages 1–8, 2012.

[72] Albert D Shieh and David F Kamm. Ensembles of one class support vector machines. In *Multiple Classifier Systems*, pages 181–190. Springer, 2009.

[73] J. R. Simmonds, G. Lawson, and A. Mayall. *A Methodology for Assessing the Radiological Consequences of RoutineReleases of Radionuclides to the Environment*. EUR, 1018-5593 ; 15760. European Commission, Directorate-General for Environment, Nuclear Safety and Civil Protection, 1995.

[74] Jerzy Stefanowski. Dealing with data difficulty factors while learning from imbalanced data. In *Challenges in Computational Statistics and Data Mining*, pages 333–363. Springer, 2016.

[75] T. J. Stocki, N. Japkowicz, I. K. Ungar, J. Hoffman, and J Yi. Summary of the data mining contest for the ieee international conference on data mining. In *Proceedings of the ICDM'08 Data Mining Contest*, 2008.

[76] T. J. Stocki, G. Li, N. Japkowicz, and R. K. Ungar. Machine learning for radioxenon event classification for the Comprehensive Nuclear-Test-Ban Treaty. *Journal of environmental radioactivity*, 101(1):68–74, 2010.

[77] J.D. Sullivan. The comprehensive test ban treaty. *Physics Today*, 51(3):24–29, 1998.

[78] David MJ Tax and Robert PW Duin. Combining one-class classifiers. In *Multiple Classifier Systems*, pages 299–308. Springer, 2001.

[79] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[80] T. Tirabassi and U. Rizza. A practical model for the dispersion of skewed puffs. *Journal of Applied Meteorology*, 34(4):989–993, 1995.

[81] D. B. Turner and J. H. Novak. User's guide for RAM. volume 1: Algorithm description and use. Technical Report EPA-600/8-78-016A, US Environmental Protection Agency, Reseach Triangle Park, NC, 1978.

[82] Byron C Wallace, Kevin Small, Carla E Brodley, Thomas Trikalinos, et al. Class imbalance, redux. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 754–763. IEEE, 2011.

[83] Defeng Wang, Daniel S Yeung, and Eric CC Tsang. Structured one-class classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(6):1283–1295, 2006.

[84] Gary M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explor. Newsl.*, 6(1):7–19, June 2004.

[85] John D Woodward, Nicholas M Orlans, and Peter T Higgins. *Biometrics:[identity assurance in the information age]*. McGraw-Hill/Osborne New York, 2003.

[86] G. Wu and E. Chang. Class-boundary alignment for imbalanced data set learning. *Proc. International Conference on Data Mining*, 2003.

[87] G. Wu and E.Y. Chang. Kba: Kernel boudnary alignment considering imbalanced datasets. *IEEE Transactions on Knowledge and Data Engineering*, 17:786–795, 2005.

[88] Roman V Yampolskiy. Human computer interaction based intrusion detection. In *Information Technology, 2007. ITNG'07. Fourth International Conference on*, pages 837–842. IEEE, 2007.

[89] Roman V Yampolskiy. Indirect human computer interaction-based biometrics for intrusion detection systems. In *Security Technology, 2007 41st Annual IEEE International Carnahan Conference on*, pages 138–145. IEEE, 2007.

[90] Roman V Yampolskiy. Motor-skill based biometrics. In *Assuring Business processes, Proceedings of the 6th Annual Security Conference, Ed. G. Dhillon. Global Publishing, Las Vegas, NV, USA*, 2007.

[91] Roman V Yampolskiy and Venu Govindraju. Behavioural biometrics : a survey and classification. 1(1), 2008.