Copyright Notices

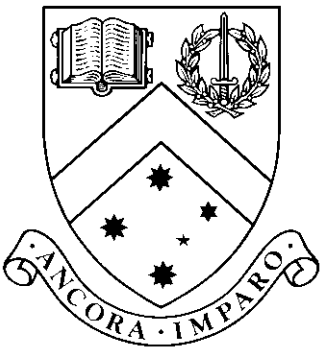
Notice 1

Notice 2

Anomaly Detection Using Isolation

by

Fei Tony Liu



Thesis

Submitted by Fei Tony Liu

for fulfillment of the Requirements for the Degree of

Doctor of Philosophy (0190)

Supervisor: A. Prof. Kai Ming Ting

Associate Supervisor: Prof. Zhi-Hua Zhou

Gippsland School of Information Technology
Monash University

June, 2011

To Jackie, Enoch, Philip and Anna

Contents

List of Tables

List of Figures

Anomaly Detection Using Isolation

Fei Tony Liu
[redacted]
Monash University, 2011

Supervisor: A. Prof. Kai Ming Ting
[redacted]
Associate Supervisor: Prof. Zhi-Hua Zhou
[redacted]

Abstract

Anomaly detection is the process of discovering unusual data patterns that are different from the majority of the data. It has been used for fraud detection in the credit card and insurance industries, as well as other applications such as intrusion detection, industrial damage detection, and medical and public health anomaly detection. Alongside predictive modelling, link analysis and cluster analysis, anomaly detection forms one of the four pillars in data mining research and applications.

Anomalies are data points that are intrinsically *few* in number and *different* from other normal data. Due to these intrinsic properties, anomalies are highly susceptible to isolation. This thesis proposes the first isolation-based anomaly detectors that detect anomalies purely based on the concept of isolation. The proposed method is fundamentally different from all existing methods that determine anomalies using distance-based or density-based approaches. Isolation-based anomaly detectors estimate the susceptibility to isolation for each data point without employing any computationally expensive distance or density measures. This fundamental allows a significantly lower processing time, higher detection accuracy and the ability to detect a wider range of anomalies, such as clustered anomalies.

This thesis explains how isolation-based anomaly detectors work in separating anomalies from the majority of data, even when there is a high volume of data. In addition, an extensive empirical evaluation and an investigation on high dimensional data are provided. Finally, we discuss possible extensions of this novel method, such as handling categorical data and data streams.

Anomaly Detection Using Isolation

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Fei Tony Liu
June 1, 2011

Acknowledgments

The insight that anomalies colligate with short path lengths in binary trees, is in itself, an anomaly. I discovered this while I was studying my studying my Master by Research course under the supervision of Kai Ming Ting and Wei Fan. Completing my master degree taught me the patience and attention to detail, which lead to the discovery of this interesting phenomenon.

For this thesis, I am deeply indebted to my supervisors Kai Ming Ting and Zhi-Hua Zhou for their constant encouragement, support, patience and excellent advices. During my candidature, I experienced many difficult circumstances, including the passing of my grandmother and a serious car accident (others at fault). I would like to thank both my supervisors for being understanding and accommodating. I would also like to thank my colleagues Swee Chuan Tan (James), Jonathan Wells, Guang-Tong Zhou and Zhoyu Fu for their in-depth discussions on various research topics. I thank Patrick Leegel for proofreading my thesis. I wholeheartedly thank my parents for their tireless efforts in bringing me up and supporting me whenever possible. I thank my loving wife Jackie for her undying support and unwavering belief in me to finish this thesis. Above all, I thank God for his generous provisions in my life and the opportunity to write this thesis. May this work glorifies His name.

The author is financially supported by Australian Postgraduate Award (APA) and Victorian ICT research scholarship during his PhD candidature.

Fei Tony Liu

Monash University

June 2011

Chapter 1

Introduction

> 'Through every rift of discovery some seeming anomaly drops out of the darkness, and falls, as a golden link, into the great chain of order.'
>
> *Edwin Hubbel Chapin*

Anomalies are data patterns that have unexpected characteristics as compared to the majority of the data. These unusual data patterns are regarded as *exceptions* in large databases and the process to discover them is referred to as anomaly detection or outlier detection. Anomaly detection has been widely adopted in medical research as well as in financial and industrial applications. Such as disease outbreak monitoring and cancer research, credit card fraud detection, industrial damage detection, and computer network traffic pattern monitoring (Aleskerov et al., 1997; Wong et al., 2003; Ertoz et al., 2004).

Anomalies are usually the minority in a data set, but their peculiar data characteristics often carry important information that leads to new discoveries or emerging trends. Contrary to the *known* normal data patterns, anomalies reveal *unknown* patterns that can arouse our curiosity to pursue the causes of the deviation; hence it enables us to develop new knowledge to explain what is previously *unknown*. As a valuable knowledge discovery tool, anomaly detection is well recognized as one of the four pillars in data mining, besides predictive modelling, link analysis and cluster analysis. Some researchers have argued that anomaly detection is closest to the motivation behind data mining as compared to the other threes (Sun, 2006).

In this thesis we introduce the first anomaly detectors, which are based on the concept of *isolation*. Isolation refers to the recursive partitioning of data until a data point is singled

out or isolated. Due to anomalies' intrinsic property of being the minority and carrying unusual data patterns, when the data are being partitioned into subgroups recursively, anomalies are more readily isolated than normal instances. My thesis statement is that:

> "Anomalies can be efficiently and effectively identified using isolation, since anomalies are more susceptible to isolation than normal instances."

The intuitive understanding of the susceptibility to isolation (or the degree of isolation) is that if a data point requires only a few partitions to be isolated, then it is highly likely to be an anomaly; however if a data point requires many partitions to be isolated, then it is likely to be a normal instance. By measuring the degree of isolation instead of the distance or density, isolation-based detectors have a significantly lower computation requirement than conventional distance- or density-based methods, which results in a drastic reduction in processing time, especially with high volume data. In the remainder of this thesis, I will attempt to prove this thesis statement. We aim to do so by investigating and evaluating the capabilities of isolation-based methods for the task of anomaly detection.

For the rest of the introduction, we shall first investigate the nature and definitions of anomalies in Section 1.1. In Section 1.2, we outline the motivations for this work and the challenges to overcome. In Section 1.3, we explain the fundamental difference between isolation-based method and existing methods, followed by the objectives and contributions of this thesis in Section 1.4. Section 1.5 details the organization of this thesis.

1.1 What are Anomalies

Finding an accurate definition to cover all aspects of anomalies remains one of the challenges in anomaly detection research. There are many ways to define anomalies in the literature, and some other synonyms being used include: outliers, deviations, novelties, exceptions, discordant observations, noise, damage, error, fault, aberration, surprise value, rogue value, peculiarity and contaminant (Chandola et al., 2009; Zhang et al., 2007). This diverse range of synonyms reflects the multifaceted nature of anomalies. While there is no universally accepted definition of anomalies, many prevalent definitions for anomalies have been given over the years, for examples:

- *"An outlier is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism."* (Hawkins, 1980)

- *"An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data."* (Barnett and Lewis, 1994)

In this thesis, *anomalies are defined as data points that are '**few**' and '**different**'* from the other data instances. These two properties are mutually dependent. Data points that are few in number but not different from other points, do not qualify as anomalies. Similarly, numerous data points that are different are also not considered to be anomalies. While these properties may be found embedded in other existing definitions, I advocate using the two properties holistically and not relying on one of them and neglecting the other. This gives rise to the isolation mechanism which incorporates these two properties into a single framework to detect anomalies.

By focusing on both the 'few' and 'different' properties concurrently, a definition of anomalies based on isolation provides a better coverage of different types of anomalies as compared to those based on conventional density or distance measures. Generally speaking, density-based methods emphasize the property of 'few'. For example, Breunig et al. (2000) define *density-based anomalies as points that lie in the lower local density with respect to the density of its local neighbourhood.* This implies that anomalies would be fewer than normal instances when they are counted within areas of the same size. On the other hand, distance-based methods emphasize the property of 'different'; Ramaswamy et al. (2000) define *distance-based anomalies as the top n points with the maximum distance to their own k^{th} neighbours.* When anomalies have a high local density compared to their neighbours, the density-based definition becomes invalid. When data sets have regions of different densities, the data in low density regions can be mistaken as anomalies, and the distance-based definition becomes invalid. On the other hand, isolation-based methods are able to detect anomalies in all of the above cases as they rely on both properties.

1.2 Motivation and Challenges

Anomaly detection is motivated by the inevitable exceptions in a data set. These exceptions can be caused by many different factors, many of which are errors in the data collection mechanism or the occurrence of uncommon events in the subject of interest. In the 80's and 90's, the motivation of anomaly detection (Hawkins, 1980; Barnett and

Lewis, 1994) centred around removing contaminants to enhance the performance of certain statistics. Stepping into the new millennium, the motivation of anomaly detection has shifted to novelty discovery as we began to realize the true value of anomalies in data. For example, anomalies in mobile phone usages could signify the fraud technique known as 'cloning' (Fawcett and Provost, 1997). The estimated cost of cloning is in the order of hundreds of millions dollars per year. Other similar discoveries could also be found in financial industries, e.g., credit card and insurance fraud detection. Apart from discovering errors or novelties, another motivation for anomaly detection is the discovery of invaluable new entities, such as new stars (Dutta et al., 2007), or new species of plants and animals (Chen et al., 2009).

This research is motivated by the increasing needs for anomaly detection in the ever-growing, mega-sized databases in modern computing. In the 21^{st} century, due to the abundant supply of data and relatively fewer anomalies, the saying of "mining needle in a haystack, so much hay and so little time" becomes a catchphrase for anomaly detection (Banerjee et al., 2008). In this statement, we catch a glimpse of the real need to provide anomaly detection for high volumes of data with a fast response time. These challenges are summarized as follows:

- **Fast detection.** The time taken in detecting anomalies is often critical in many real world applications, for example, network intrusion detection or credit card fraud detection. The extent of the damages or financial losses depend on the speed of detection and any remedial actions that may be taken, thus a fast detection is crucial to minimize losses.

- **High volume data.** This challenge is specific to the efficient use of computer memory. Many algorithms are required to hold the entire or a large proportion of data in the memory. Some even require a working space larger than the data size. When the size of the data becomes very large, anomaly detectors fail due to an unattainable memory requirement.

- **High detection accuracy.** An anomaly detector is useless if it produces too many false positives and the users are required to go through the data manually. Therefore, a high detection accuracy is essential to a good performing anomaly detector.

- **Masking and Swarming effects** (Hawkins, 1980). These are two well-known challenges for existing anomaly detectors. Masking occurs when some anomalies are not detected due to the presence of other anomalies. Swarming occurs when some normal instances are mistaken to be anomalies due the presence of other anomalies. It is thus desirable if a new detector is able to avoid these effects.

- In **high dimensional data**, the instances are sparse points in the input space (Beyer et al., 1999). It is therefore difficult to distinguish anomalies from the normal points in high dimensional space where any two points have similar inter-distance.

These challenges in anomaly detection have motivated many researchers to look for better approaches to cater for the growing demands of anomaly detection. As the traditional statistical approaches usually carry a $O(n^2)$ time complexity and place assumptions on data distribution, researchers are driven to look for alternatives in the areas of Machine Learning and Data Mining.

In the field of Machine Learning and Data Mining, it is commonly believed that "One of the rules of thumb for good anomaly detection methods is that they are able to represent the data well" (Williams et al., 2002). A number of researchers extend existing data mining algorithms to anomaly detection from their original usage in predictive modelling. For examples, k-NN (Knorr and Ng, 1998; Wettschereck, 1994; Ramaswamy et al., 2000), neural network (Williams et al., 2002; Bishop, 1994) and SVM (Tax and Duin, 2004) have all been used for anomaly detection. The advantage of such an adaptation is that the researchers and experts are able to extend their existing knowledge and understanding to anomaly detection tasks. However, these extensions are not without their problems. For example, the k-NN algorithm suffers from the masking effect since only the nearest neighbours of a point are considered. Because these methods were not specifically designed for anomaly detection, their definitions or assumptions often do not suit the task of anomaly detection. In addition, most of them are computationally expensive. This situation has motivated us to develop an isolation-based method, which has been primarily designed for the task of anomaly detection, bearing in mind the challenges mentioned above.

1.3 Isolation-Based Methods are Fundamentally Different

The use of isolation is a paradigm shift from the conventional distance and density measures. It is essentially a different way of thinking that enables isolation-based detectors to tackle the existing challenges more efficiently and effectively, since they are no longer subject to the limitations faced by distance- and density-based methods. The conventional methods profile the normal instances first and then use the profile to detect anomalies, those points falling outside the profile are deemed to be anomalies. This is an indirect approach, since the primary target in anomaly detection are the anomalies, not the normal instances. Existing anomaly detectors are effectively asking *two* indirect questions to obtain the answer: 1. *How to model normal instances?* 2. *Does this particular instance fit the model? If not, it must be an anomaly.* This is a logical elimination approach, however, the normal instances form the majority of data and therefore the workload for the modelling depends heavily on the data size. When the data size grows, the workload in profiling the normal instances increases dramatically. This means the anomaly detectors are unable to scale up for very large databases without significantly compromising the processing time. It is also questionable whether directing most of the resources to profile the normal instances is a productive course of action.

The isolation-based method is fundamentally different. Isolation-based detectors set their goal as detecting the anomalies directly through the recursive partitioning to isolate every data point. When a particular data point is isolated with a few partitions, it is likely to be an anomaly; if the data point is not isolated within a certain rounds of partitioning, it is likely to be a normal instance and the recursive partitioning is terminated. In this case, the anomaly detectors are effectively asking *one* direct question: *Is this particular instance likely to be an anomaly?* In asking this direct question, an instance can be determined to be an anomaly or normal instance within a few rounds of partitioning and no further resources need be wasted on that instance. Thus isolation-based detectors are able to focus all their computational resources on the task of detecting the anomalies, which in turn keeps the processing time low. Since it is unnecessary to store the profile information of the normal instances, the memory requirement is small.

The above-mentioned fundamental difference is the factor that enables isolation-based methods to overcome the limitations of the conventional methods. We will further discuss the differences between the isolation-based methods and the other methods in Chapter 2.

1.4 Contributions

In this thesis, we aim to investigate whether anomaly detection is achievable with an isolation framework, which encapsulates the two interdependent anomaly properties of 'few' and 'different'. The property 'few' is usually measured by the quantity of the data points and the property 'different' is usually measured by the distances between the data points. The difficulty lies in combining these two properties, which lead us to form the following five research objectives to guide our research:

1. To establish an Isolation framework which is able to measure or estimate the degree of isolation of each data point. We also investigate the strengths and weaknesses of this Isolation framework.

2. To provide both empirical and analytical evidences that the isolation-based methods are more capable of performing anomaly detection than the existing methods. To examine their performance in terms of processing time, high volume data and detection accuracy.

3. To examine how well the isolation-based methods deal with different types of anomalies. We also examine the masking and swarming effects caused by the various types of anomalies. Most anomaly detectors only perform well with scattered anomalies and very poorly with clustered anomalies.

4. Since our proposed isolation framework utilizes binary trees to construct its models and the binary tree is well understood and researched in the literature, it is important to provide a theoretical explanation on the role of binary tree in the isolation framework.

5. To investigate ways to detect anomalies in the high dimensional spaces.

As a result of these five objectives, the contributions of this thesis are as follows:

- This thesis proposes the first isolation-based anomaly methods, which are fundamentally different from conventional methods.
- It is shown that isolation-based methods are able to detect anomalies efficiently with a low linear-time complexity and a small memory requirement.
- The characteristics of the isolation-based anomaly methods are illustrated and explained.
- A detailed empirical evaluation on the proposed methods is provided.
- A solution for detecting local clustered anomalies is proposed.
- The behaviours of isolation-based methods on the high-dimensional data are studied.

1.5 Organization

The rest of this thesis is organized as follows:

In Chapter 2, we review related developments in the field of anomaly detection. We first discuss the three common approaches in unsupervised learning, namely the distance-based approach, density-based approach and model-based approach. Then we explain the development in handling the high-dimensional data.

Chapter 3 introduces the basic concept of the isolation-based approach. We first illustrate the basic characteristic of isolation. Then we demonstrate situations in which the proposed approach has advantages over the existing density- and distance-based approaches. Next, we show that using a small sub-sampling size to build an isolation model yields a higher detection performance in some situations.

Chapter 4 details the algorithmic framework of the proposed methods. We also include a formulation of the deterministic split-point selection criterion and random hyper-plane. We propose two isolation-based methods, iForest and SCiForest, which have distinct characteristics in detecting anomalies.

Chapter 5 highlights the different characteristics of the two proposed methods, including their performance with clustered anomalies, breakdown analyses and their performance in detecting the anomalies among arbitrary patterns. We will also examine the sensitivity of their different parameters in terms of the detection performance.

Chapter 6 provides an empirical evaluation of the proposed methods. Using twelve natural data sets to evaluate the anomaly detectors, our evaluation examines the detection performance and the efficiency of the proposed methods compared to the other state-of-the-art anomaly detectors.

Chapter 7 examines the behaviours of the proposed methods in dealing with high dimensional data. Chapter 8 provides insight into possible future work, including anomaly detection in data streams, handling categorical data and better handling of the high dimensional data. Chapter 9 concludes this thesis.

Chapter 2

Literature review

'All generalizations are false, including this one.'

Mark Twain

In general, there are three types of anomalies: point anomalies (Chandola et al., 2009), contextual (or conditional) anomalies (Song et al., 2007) and collective anomalies (D'haeseleer et al., 1996; Sun et al., 2007).

Point anomalies refer to data instances being anomalous with respect to the rest of the data, which are the most commonly studied type of anomaly. For example, given a place in tropical climate, e.g. the Fiji Islands, a temperature of below 0°C would be an anomaly at any time of year.

Contextual anomalies are data instances which have exceptional behaviours in a given context and the same data instances can be perfectly normal in other contexts. For example, a temperature of 45°C in summer in some parts of Australia is considered normal, however, if this were to happen in winter; it would be considered abnormal.

Collective anomalies are abnormal due to an unusual collection of data sequences. Individual data instances in a data sequence may not be abnormal if they appeared in other combinations of sequences. The combination of the data instances is the determining factor of these anomalies. Collective anomalies apply to time-series data. Illustrative examples of collective anomalies can be found in (Goldberger et al., 2000) in the context of the human electrocardiogram outputs. Unusual patterns in these outputs are collective anomalies as compared to the regular patterns. This thesis focuses on the point anomalies.

For the rest of this chapter, in Section 2.1, we first briefly describe the three major categories of anomaly detection, which are the supervised learning, unsupervised learning and semi-supervised learning. Then, in Section 2.2, we concentrate on the unsupervised learning and present three common approaches in detecting anomalies. Finally, in Sections 2.3 and 2.4, we provide the research development in handling high dimensional data and discuss the distinctiveness of our isolation-based approach.

2.1 The Three Categories of Anomaly Detection

Anomaly detection can be divided into three major categories: supervised learning, semi-supervised learning, and unsupervised learning depending on the availability of the labelled data. This thesis focuses on unsupervised learning as it does not assume the availability of labelled data.

In the supervised learning, the ground truth of both the normal points and anomalies are used to train a model; then the model is applied to differentiate the anomalies from the normal points. The typical approach is to apply a classification model to learn the difference between the normal points and anomalies. The major drawback of the supervised learning is that it is often very expensive to obtain accurately labelled data (Zhang et al., 2007). Since the anomalies are the minority, the learned models may be poor at detecting unseen anomalies. To resolve this issue, many researchers have proposed injecting artificial anomalies in the training set to improve the predictive performance of the models (Fan et al., 2001; Steinwart et al., 2005). The artificial anomalies are usually randomly generated so that the detection models generalize well against any unseen anomalies. Other issues in the supervised anomaly detection are the same as those in learning imbalanced classes in classification models and they are being addressed by (Joshi et al., 2001, 2002; Vilalta and Ma, 2002; Phua et al., 2004; Chawla et al., 2004). The issues in learning imbalanced classes include: i) the anomalies are under-represented, ii) misclassifying anomalies can be more costly than normal instances and iii) the distribution of the test data can be very different from the training data.

In the semi-supervised learning, the anomaly detectors are given only the labels for one of the two classes (anomalies/normal points). Most semi-supervised learners learn from the labels of the normal points; due to the high cost of labelling the anomalies. The

detectors that assume the availability of the anomaly class (Dasgupta and Majumdar, 2002; Dasgupta and Nino, 2000) are not as popular. The semi-supervised learners also suffer a similar drawback as the supervised learners in that a representative set of the normal instances is difficult to obtain in real-life applications (Zhang et al., 2007).

Unsupervised anomaly detection does not assume the availability of any class labels. Under such an assumption, the learning algorithm is required to detect the anomalies using information provided within the data, e.g., the distance-based methods identify the anomalies using the distances between a point and its nearest neighbours. Typically unsupervised learning can be further divided into three approaches, a) Distance-Based, b) Density-Based and c) Model-Based (Tan et al., 2005). There are overlaps between these approaches and some existing methods fit well into multiple approaches.

In terms of the output of the anomaly detection, we are only interested in anomaly detectors that output a score reflecting the degree of the anomaly rather than a label describing whether a point is anomalous or not. A score can be converted into a label by a given threshold value.

During the course of the following discussion, we will be using the terms "scattered", "clustered", "global" and "local" to describe different types of anomalies and their definitions are as follows:

- **Scattered anomalies** are the anomalies scattered outside the range of the normal points.
- **Clustered anomalies** are the anomalies which form clusters outside of the range of the normal points.
- **Global anomalies** are anomalous data points with respect to the other points in the entire data set. However, they may not be anomalous with respect to their local neighbourhoods (Sun, 2006).
- **Local anomalies** are data points that are significantly different from their local neighbourhoods. However, they may not be considered anomalies with respect to the majority data points in the data set (Sun, 2006).

Clustered anomalies are the most challenging among these anomalies. Detecting clustered anomalies is largely related to a problem known as the 'masking effect' (Murphy, 1951;

Barnett and Lewis, 1994; Pearson, 2005). Examples of clustered anomalies can be found in industrial manufacturing process (Chapter 3 in Pearson (2005)) and in spatial and temporal domains (Janeja and Atluri, 2008). Apart from these general descriptions of the different approaches to detecting anomalies in unsupervised settings, we will also discuss why clustered anomalies are challenging for the different approaches in the following section.

2.2 Unsupervised Anomaly Detectors

2.2.1 Distance-based approach

> **Distance-based methods** identify anomalies as data points that are distant from their neighbours.

Distance-based methods are also known as proximity-based techniques (Hodge and Austin, 2004). Examples of distance-based methods are $DB(p, D)$ (Knorr and Ng, 1998; Knorr et al., 2000), D_n^k (Ramaswamy et al., 2000), ORCA (Bay and Schwabacher, 2003), SNIF (Tao et al., 2006) and DOLPHIN (Angiulli and Fassetti, 2009).

Knorr et al. (2000) and Knorr and Ng (1998) define "*a point* $\mathbf{x}$ *in a data set* X *is a* $DB(g, D)$ *outlier if at least a fraction* g *of the points in* X *lies at a greater distance than* D *from* $\mathbf{x}$". Their aim was to speed up the distance calculations and proposed three algorithms, i.e., index-based, nested-loop and cell-based algorithms. The index-based algorithm utilizes a standard multi-dimensional indexing structure to search for the nearest neighbours within a certain range. Once all the anomalies are detected, the search is terminated. Without using an index, the nested-loop algorithm uses a blocked oriented nested loop to compute a count of neighbours for each point. The counting for one point stops when the number of neighbours exceeds a threshold. The cell-based algorithm makes use of the properties of cells to eliminate non-anomalies in order to speed up the search operation. The first two algorithms have an $O(n^2 d)$ time-complexity, the last one has an $O(c^d + n)$ time-complexity, where c is a constant. $DB(p, D)$ is sensitive to the parameters D and p, and does not provide a ranking of the anomalies. Distance-based methods can be inefficient, e.g., the nested-loop algorithm. In a nested-loop algorithm, for each instance $\mathbf{x}$ in a data set X, one scan of the entire data set is necessary to verify whether $\mathbf{x}$ is a normal instance or an anomaly. For a normal instance, the scan can be terminated when k

neighbours are found within D distance from $\mathbf{x}$. However, for an anomaly, complete data scans of the entire data set are necessary. Multiple complete data scans are not required in the isolation-based methods proposed in this thesis, which makes them more efficient in terms of the processing time. Some distance-based methods have linear runtime $O(n)$, we will compare isolation-based methods with the state-of-the-art distance-based method in Chapter 6 including the scalability of processing time.

Ramaswamy et al. (2000) modify the definition by Knorr et al. (2000) based on the distance of k^{th} nearest neighbour. This provides a ranking for a predefined number of anomalies and simplifies the user-defined parameter to a single k. Ramaswamy et al. (2000) also optimize the index-based and nested-loop algorithms and introduce a partition-based algorithm which prunes off unnecessary distance calculations.

ORCA (Bay and Schwabacher, 2003) speeds up the nested-loop algorithm and has a near linear time complexity, provided that the sequence of instances is sufficiently randomized. ORCA randomizes the sequence of instances and partitions them into blocks. ORCA keeps track of a set of data points as potential anomalies along with their anomaly scores. The minimum anomaly score of the set is used as a cut-off. The cut-off is updated if a point with a higher score is found. If a point has a lower score than the cut-off, the point is pruned. This pruning process speeds up the distance calculations only if the order of the data is uncorrelated. ORCA's worse case time-complexity is still $O(n^2)$ and the I/O cost is quadratic (Tao et al., 2006).

SNIF (Tao et al., 2006) reduces the I/O cost with a trade-off for a higher memory requirement. Using a prioritized flushing technique, SNIF is able to reduce the number of data scans to two or three times. When reading data instances into the memory, the prioritized flushing prioritizes the data points according to a priority scheme. At the critical moment (when the memory is full), half of the data instances are removed according to their priorities. This process continues until all the anomalies and normal points are marked.

DOLPHIN (Angiulli and Fassetti, 2009) gains efficiency by combining three strategies: 1) a policy to select data objects in the main memory, 2) a pruning rule and 3) a similarity search. DOLPHIN has a time-complexity of $O(\frac{k}{p}nd)$, where p is the probability of randomly picking a point from a data set that is a neighbour of the point in the index.

With exactly two data scans, DOLPHIN utilizes a $\frac{k}{p}$ amount of memory. However, DOLPHIN degenerates to $O(n^2d)$ when $\frac{k}{p}$ is as large as n. Empirically, DOLPHIN compares favourably to ORCA and SNIF in terms of the execution time and I/O cost (Angiulli and Fassetti, 2009).

Strengths and weaknesses. Distance-based methods are simple and straight-forward, and many different schemes can be employed to speed-up the processing time. However, applying distance-based methods to large data sets can be very expensive and speed-ups are limited to low dimensional data since every data point appears sparse in high dimensional space (Beyer et al., 1999). On the other hand, since distance is measured uniformly across the entire data set, a distance-based approach is not able to handle data sets with regions of different densities. For example, normal instances in low density regions would be mistaken as anomalies because their inter-distances are long compared to normal instances in high density regions. Similarly, clustered anomalies would be mistaken as normal points as clustered anomalies have short inter-distances among themselves.

The development of distance-based methods has mainly been in a number of improvements to speed up the distance calculations with little or no change to their anomaly definitions. As a result, distance-based methods would still have trouble handling data with regions of different densities. Distance-based methods which rely on the full dimensional pair-wise distance calculations are subject to the curse of dimensionality for high dimensional data.

2.2.2 Density-based approach

> **Density-based methods** operate on the principle that the instances in low density regions are considered anomalies.

The density is usually defined in terms of distance and therefore the density-based approach is closely related to the distance-based approach, especially in $k$nn-based methods. For example, the density is defined by the number of points within a given radius. The known limitations of this density definition are that i) it cannot correctly identify anomalies in data with regions of different densities and ii) it is sensitive to the setting of parameters (Tan et al., 2005). To rectify the first limitation, the notion of the relative density was introduced (Breunig et al., 2000), which defines normal instances as having a

density similar to those of their neighbours; and anomalies as having a lower density as compared to their neighbours.

Some well-known density-based methods are: *Local Outlier Factor* (LOF) (Breunig et al., 2000), *Connectivity-based Outlier Factor* (COF) (Tang et al., 2002), *LOcal Correlation Integral* (LOCI) (Papadimitriou et al., 2003) and Resolution-based Outlier Factor (ROF) (Fan et al., 2006).

In LOF (Breunig et al., 2000), a LOF value is computed for each instance, which is the inverse of the relative density. The LOF value indicates the sparseness of a point in relation to its local neighbourhood. The points having the largest LOF values are considered to be anomalies. LOF is defined as:

$$LOF_k(\mathbf{x}_a) = \frac{\sum_{\mathbf{x}_b \in N_k(\mathbf{x}_a)} \frac{lrd(\mathbf{x}_b)}{lrd(\mathbf{x}_a)}}{|N_k(\mathbf{x}_a)|},$$

where $N_k(\mathbf{x})$ is the set of the k nearest neighbours of $\mathbf{x}$ and $lrd(\mathbf{x})$ is the local reachability density of $\mathbf{x}$. $lrd(\mathbf{x})$ is the inverse of the average reachability distance of the instance x from its neighbours. As the detection performance of LOF is sensitive to the choice of k used in the underlying $k$nn algorithm, the standard LOF algorithm searches through a range of k values and uses the maximum LOF value as the anomaly score for a given point. This search process is very expensive.

COF (Tang et al., 2002) enhances LOF by taking into account the connectivity of the neighbours of a point, in addition to the density of its neighbours. A point is connected to another point (connectivity) if their inter-distance is less than a certain threshold, otherwise, they are not connected (disconnectivity). The anomaly score is calculated using the ratio of the average distance from a point to its k-distance neighbours and the average distance of its k-distance neighbours to their own k-distance neighbours. The k-distance neighbours of a point are other points that fall within a radius of k. Tang et al. (2002) separate the treatment of the low-density points and isolated points. COF defines isolativity as the degree of disconnectivity to other neighbouring points. This isolativity is different from the isolation framework proposed in this thesis. Isolativity is defined by density and the isolation framework is defined by the path length of binary trees.

Papadimitriou et al. (2003) propose LOCI, which is based on the Multi-Granularity Deviation Factor (MDEF). MDEF measures the relative deviation of density of a point's

neighbourhood in order to cope with clustered anomalies. For each instance $\mathbf{x}$, LOCI first identifies a region defined by (user supplied) radius r from $\mathbf{x}$ and counts the number of its nearest neighbours n, within the region. Then, for each neighbour of $\mathbf{x}$ and including $\mathbf{x}$, LOCI counts the number of instances within a smaller region of radius αr, centred at each neighbour, where $\alpha < 1$. These counts are averaged to produce $\hat{n}$. MDEF is defined as the relative difference between $\hat{n}$ and n, i.e., $(\hat{n} - n)/\hat{n}$. A normal instance will have a MDEF close to 0, because it has similar density to its surrounding. An anomaly will have a MDEF much greater than 0, since it has a different density than its surroundings.

A grid-based variant aLOCI has a time complexity of $O(nLdg)$ for building a quad tree, and $O(nL(dg+2^d))$ for the scoring and flagging, where L is the total number of levels and $10 \leq g \leq 30$. LOCI is able to detect clustered anomalies, however, detecting the anomalies is not straight-forward and requires a manual interpretation of the LOCI curve for each point. LOCI is more computationally intensive compared to the isolation-based detectors as it requires distance calculations for defining regions.

ROF (Fan et al., 2006) defines an anomaly as a point which is inconsistent with the majority of the data at different resolutions. ROF is defined as the accumulated ratio of the sizes of clusters containing a point in two consecutive resolutions. The anomalies are the data points with the lowest ROF values.

K-d Tree (Chaudhary et al., 2002) is a tree-based density model, which partitions the instance space into rectangular regions. Each region has a nearly uniform sparseness and the degree of anomaly of a region is measured by its relative sparseness to its neighbouring regions. Since sparseness is the inverse of density, this method requires density calculations for each region. In contrast, our proposed isolation-based method uses the path length instead of the density to detect anomalies.

Strengths and weaknesses. As in the distance-based approach, the density-based approach is free from any assumptions regarding the data distribution. However, the density-based approach is able to identify any local outliers in data with varying densities. The density-based approach is sensitive to the choice of parameters and has a high time complexity of $O(n^2)$ before any optimization. Similar to the distance-based approach, the density-based approach depends on a full dimensional distance computation and hence is subject to the curse of dimensionality. We also notice that for $k$nn based methods to deal

with clustered anomalies, k would have to be larger than the size of a group of anomalies for successful detection. Increasing k increases the processing time, and the size of a group of clustered anomalies can be quite large in some real life applications. It makes $k$nn based density methods unworkable for those real life applications. The development of density-based methods is marked by an increasing number of derivatives from the original notion of density. Density-based methods introduce more and more sophisticated definitions to handle data of varying densities and anomalies of different types. Since density estimation is usually costly, efficiency is never the strength of density-based methods. The development of density-based methods stagnates due to these efficiency issues.

2.2.3 Model-based approach

> Model-based methods construct a model from the data, then identify anomalies as data points that do not fit the model well.

Examples of model-based approach are: classification-based methods (Abe et al., 2006), Replicator Neural Network (RNN) (Williams et al., 2002), one-class SVM (Tax and Duin, 2004), clustering-based methods (He et al., 2003), link-based methods (Ghoting et al., 2004), Convex peeling (Rousseeuw and Leroy, 1987), the minimum volume ellipsoid (MVE) and the minimum covariance determinant (MCD) (Rousseeuw, 1984).

In classification-based methods (Abe et al., 2006), when an anomaly class is not known, the general approach is to first convert an unsupervised anomaly detection problem into a supervised classification problem by injecting instances of a "background" class, in complement to the normal class. These instances are randomly generated so that a decision boundary between the normal class and background class can be learned. By utilizing their generalization ability, classifiers are able to detect anomalies as instances of the background class. Random Forests (Shi and Horvath, 2006) is also a classification-based method and utilizes a proximity matrix in addition to a classifier to detect anomalies. When two data points end up in the same leaf of a tree, the corresponding cell of these two points in the proximity matrix is incremented by one. Random Forests has an $O(n^2)$ space complexity because the proximity matrix is of size $n \times n$, which limits its applications to small data sets only.

In RNN (Williams et al., 2002), a multi-layer perceptron neural network with three hidden layers is employed to replicate the input variables. The anomaly detection ability

comes from the fact that anomalies are poorly reconstructed using these replicator neural networks. So, data points that are poorly reconstructed are deemed anomalies. However, the conditions under which the data points would be poorly reconstructed could not be fully modelled.

One-class SVM (Tax and Duin, 2004) aims to find the smallest region that contains most of the normal data points; and points outside of this region are then deemed anomalies. The detection of one-class SVM is sensitive to the choice of kernels and parameters. It is impossible to find out beforehand which kernel and parameter setting are best suited for a given data set. Therefore, a number of time consuming trials of different kernels and parameter settings are necessary to find a good setting for a given data set.

In clustering-based methods (He et al., 2003), the data are first clustered by a clustering algorithm; then, any small clusters or points that are distant from the large cluster centroids are deemed anomalies. The performance of clustering-based methods varies depending on their underlying clustering algorithms. Since many clustering algorithms are based on distance and density measures, they are also subject the limitations of these measures.

LOADED (Ghoting et al., 2004) employs the idea from link analysis that similar data have more "supports" and anomalies are those that have less. An anomaly score in LOADED is calculated using the number of common attribute-value pairs shared by two data instances. For numeric attributes, a correlation matrix is used to estimate the level of agreement in addition to the score calculated for categorical attributes. It is one of the few methods that can handle mixed type data. However, the calculation of the supports and correlation matrix is very expensive for large data sets. The computational complexity of LOADED is $O(n \times d^2)$, where d is the number of continuous attributes.

Convex peeling (Rousseeuw and Leroy, 1987) is a depth-based method using ideas from computational geometry. Using the definition of the half-space depth (Tukey, 1977), each data point is assigned a depth and the anomalies are points that have lower depth values. Convex peeling is only suitable for up to 2-dimensional data. In general, depth-based methods measure how deep a point is with reference to a single data cloud (Liu et al., 1999). In contrast, the isolation-based approach measures how isolated a point is without any assumption about the data distribution and hence is able to handle data with multiple data clouds.

OutRank (Moonesignhe and Tan, 2006) is a stochastic graph-based algorithm, which detects both scattered and clustered anomalies (or small clusters of outliers) using the concept of random walk. In OutRank, first, a weighted graph is built by instance similarity or the number of shared neighbours between instances. Then, a Markov chain model is built upon this graph for assigning an anomaly score to each instance. Although OutRank is able to detect clustered anomalies, the computation of the weighted graph requires a significant amount of computing resources creating a bottleneck for use in real life applications. The computational complexity of OutRank is $O(n^2)$.

Statistical methods seek a subset of the entire data set for which some criteria are minimized. For examples, the minimum volume ellipsoid (MVE) (Rousseeuw, 1984) performs by finding the minimum volume ellipsoid that covers all h points sampled from the entire data population. The final ellipsoid is established after several trials. The minimum covariance determinant (MCD) (Rousseeuw, 1984) functions by finding the h points whose covariance matrix has the lowest determinant. In these two methods, once a model of the normal points is established, any point that deviates from this model is deemed an anomaly. Statistical methods usually carry strong assumptions on the data distribution and are generally only useful for univariate or low dimensional data sets.

Strengths and weaknesses. Most of the model-based methods are ported from existing data mining techniques or based on standard statistical techniques. They are well studied, well understood and are well suited for data with known properties. Their detection performances are very much dependent on how well the data fit into their model assumptions. Often, the detection performances are good, only when their anomaly definitions suit a particular data set. Except for the statistical methods, the development of model-based methods is often ad hoc. In detecting clustered anomalies, most model-based methods are designed to detect only scattered anomalies, except for clustering based algorithms and OutRank.

2.3 High Dimensional Data

In high dimensional space, distance measures become meaningless (Beyer et al., 1999) and every data point appears as a sparse point, thus it becomes impossible to identify the

anomalies. This phenomenon is referred to as the curse of dimensionality. To keep using distance-based methods, many dimension reduction schemes have been proposed. For examples, Aggarwal and Yu (2001) proposed using evolutionary algorithms to find a low dimensional projection of a given high dimensional data set with low sparsity coefficients. In these low dimensional projections, the anomalies can be exposed. The reduced dimensionality allows distance- or density-based methods to detect anomalies without the effects of a high-dimensionality. Angiulli and Pizzuti (2005) proposed HilOut, which identifies the anomalies as the most 'weighted' points in a Hilbert space. Weight for each point is defined as the sum of distances from its nearest neighbours. Filzmoser et al. (2008) proposed an algorithm using principal components to identify anomalies in a transformed data space. In Chapter 7, we will look at ways in which isolation-based methods are able to incorporate low-cost dimension reduction schemes to handle high dimensional data.

2.4 Distinctiveness of Isolation-Based Approach

To highlight the distinctiveness of the isolation-based approach, we describe its differences from other approaches in the following paragraphs.

The isolation-based approach is different from distance-based methods in that (i) isolation-based methods measure the susceptibility of a point to being isolated, rather than the difference between two data points, (ii) the isolation is measured at different neighbourhood levels, and (iii) isolation-based methods are not required to perform multiple complete data scans on a given data set.

The main difference between density-based and isolation-based approaches is the way they measure the degree of anomaly. While the density is usually measured locally with respect to the nearest neighbours of a data point, isolation is naturally measured across different granularities beyond the immediate local neighbourhood of a data point. This difference translates into the ability of isolation-based methods to detect a wider range of anomalies as described in Chapter 6.

Besides the way they measure the degree of anomaly, an isolation-based method is also free from the full dimension distance calculations as required in density- and distance-based approaches. This makes an isolation-based method a more efficient choice compared to distance- and density-based approaches as shown in Chapter 6. Also, isolation-based

methods are affected differently under an increase of dimensionality. Details can be found in Chapter 7.

The isolation-based methods can be considered model-based methods, however, isolation-based methods do not explicitly construct a model to profile normal instances and isolation-based methods are specifically designed for anomaly detection. Unlike many model-based methods, e.g., MVE, MCD and most depth-based methods, isolation-based methods do not have an assumption of the data distribution or unimodality. This gives isolation-based methods the flexibility to handle a wider range of data distributions. The goal of a model-based method is to construct a model of the normal instances; whereas the goal of an isolation-based method is to measure the degree of isolation of each data point.

Apart from these, the characteristics of the isolation trees enable isolation-based methods to exploit sub-sampling to an extent that is not feasible in existing methods (more details are provided in Section 6.3). Because of that, isolation-based methods have low linear time complexities with very small memory requirements; and with a fixed sub-sampling size, they have constant training time and space complexities. To our best knowledge, the best performing method achieves only approximate linear time and space complexities (Angiulli and Fassetti, 2009). Also, isolation-based methods are able to handle extremely large data sets (more details are provided in Section 6.3).

Chapter 3

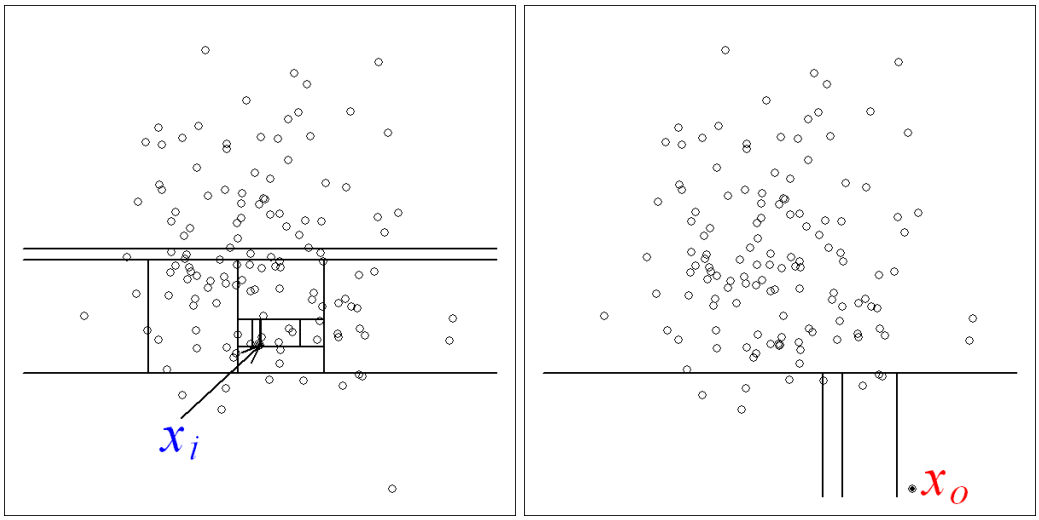
Basic Concepts of Isolation

'One person's noise is another person's signal.'

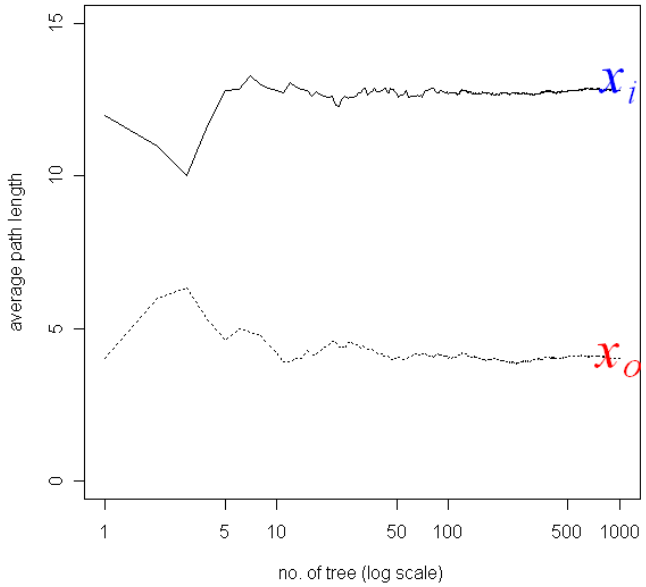
Anonymous

In this thesis, the term *isolation* means 'the act of separating an instance from the rest of the instances'. In general, an isolation-based method measures an individual instance's susceptibility to being isolated; and the anomalies are those that have the highest susceptibility. To realize this idea of isolation, we turn to a data structure that naturally isolates data. In randomly generated binary trees where the instances are recursively partitioned, these trees produce noticeably shorter paths for anomalies since (a) in the regions of anomalies, the fewer instances result in a smaller number of partitions — a shorter path length in the tree, and (b) instances with distinguishable attribute-values are more likely to be separated early in the partitioning process. Hence, when a forest of random trees collectively produces shorter path lengths for a set of points, they are likely to be anomalies. Throughout this chapter, we exemplify isolation-based methods using random trees called Isolation Forests or iForest. This is used in all illustrations and performance results reported in this chapter. The details of iForest's implementation can be found in Section 4.6 in Chapter 4.

Consider a random sample of n observations from some d-variate distribution, $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, $\mathbf{x}_j = [x_j^1, ..., x_j^d]$. X are random vectors in $\mathbb{R}^d$. The anomaly detection problem is in finding anomalies $\mathbf{x}_o \in X$, that fulfil certain criterion of being anomalies. In the case of isolation-based methods, the criterion is being "few" and "different" as mentioned in



(a) Isolating $\mathbf{x}_i$ (b) Isolating $\mathbf{x}_o$



(c) Average path lengths converge

Figure 3.1: Anomalies are more susceptible to isolation and hence have short path lengths. Given a Gaussian distribution (135 points), (**a**) a normal point $\mathbf{x}_i$ requires twelve random partitions to be isolated; (**b**) an anomaly $\mathbf{x}_o$ requires only four partitions to be isolated. (**c**) averaged path lengths of $\mathbf{x}_i$ and $\mathbf{x}_o$ converge when the number of trees increases.

Chapter 1.1. Using iForest, these criteria imply that the anomalies are data points with short path lengths.

To demonstrate how anomalies are more susceptible to isolation, we illustrate an example in Figures 3.1(a) and (b) to visualise the random partitioning of a normal point $\mathbf{x}_i$ versus an anomaly $\mathbf{x}_o$. We observe that a normal point, $\mathbf{x}_i$, generally requires more partitions to be isolated. The opposite is true for an anomaly, $\mathbf{x}_o$, which generally requires

less partitions to be isolated. In this example, the partitions are generated by randomly selecting an attribute r and then randomly selecting a split value p between the maximum and minimum values of the selected attribute r. Since recursive partitioning can be represented by a tree structure, the number of partitions required to isolate a point is equivalent to the path length from the root node to the terminating node. In this example, the path length of $\mathbf{x}_i$ is greater than the path length of $\mathbf{x}_o$.

Since each partition is randomly generated, the individual trees are generated with different sets of partitions. We average the path lengths over a number of trees to find the expected path length. Figure 3.1(c) shows that the average path lengths of $\mathbf{x}_o$ and $\mathbf{x}_i$ converge when the number of trees increases. Using 1000 trees, the average path lengths of $\mathbf{x}_o$ and $\mathbf{x}_i$ converge to 4.02 and 12.82 respectively. They show that anomalies have path lengths shorter than those of normal instances.

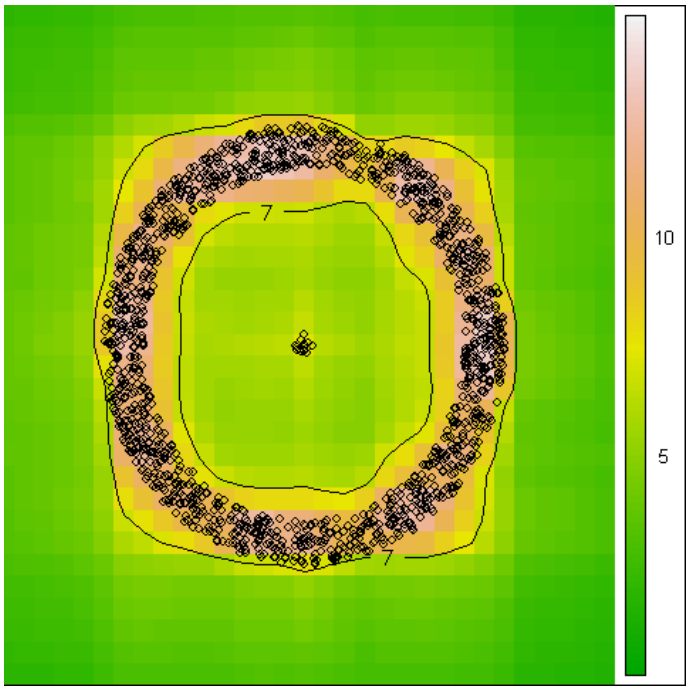


Figure 3.2: An isolation-based method is able to detect not only outlying scattered points, it can also detect anomalies surrounded by normal points as shown above. Thirteen anomalies are separated from the surrounding normal points by path length (< 7). The area with path length less than 7 is indicated by green background colour.

In addition to detecting scattered anomalies as shown above, isolation-based methods are also capable of detecting anomalies surrounded by normal points. Figure 3.2 illustrates such a scenario in which the normal points form a ring shape of which the anomalies are at the 'centre'. Using an isolation-based method, a path length contour with two contour

lines of 7 are provided in the figure. The anomalies have path length below 7 and the normal points above 7.

Definition 1. ***Isolation Tree (iTree).*** *Let T be a node of an isolation tree. T is either an external-node with no child, or an internal-node with one test and two daughter nodes (T_l, T_r). A test consists of an attribute r and a split value p such that the test $(r < p)$ divides the data points into T_l and T_r.*

Given a sample X' of ψ instances from X, to build an isolation tree (iTree), we recursively divide X' by randomly selecting an attribute r and a split value p at each internal node, until either (i) the node has only one instance or (ii) all data at the node have the same value. An Isolation Tree or iTree is a *proper binary tree*, where each node in the tree has zero or two daughter nodes. Assuming all instances are distinct, each instance is isolated on an external node when the iTree is fully grown, in which case the number of external nodes is ψ and the number of internal nodes is $\psi - 1$; the total number of nodes of an iTree is $2\psi - 1$; and thus the memory requirement is bounded and grows linearly with ψ.

The task of anomaly detection is to provide a ranking that reflects the degree of anomaly. The way to detect anomalies using iTrees is to sort the data points according to their average path lengths in ascending order; the anomalies are then points that are ranked at the top of the list. We define the path length as follows:

Definition 2. *: **Path Length** $h(\mathbf{x})$ of a point $\mathbf{x}$ is the number of edges for $\mathbf{x}$ to traverse through an iTree from the root node until the traversal is terminated at an external node.*

We employ the path length as a measure of the degree of susceptibility to isolation:

- a short path length indicates a high susceptibility to isolation,
- a long path length indicates a low susceptibility to isolation.

In order to further understand this susceptibility, we provide an analysis in Appendix B, which explains how a property of isolation trees gives shorter path lengths to anomalies.

3.1 Isolation, Density and Distance measures

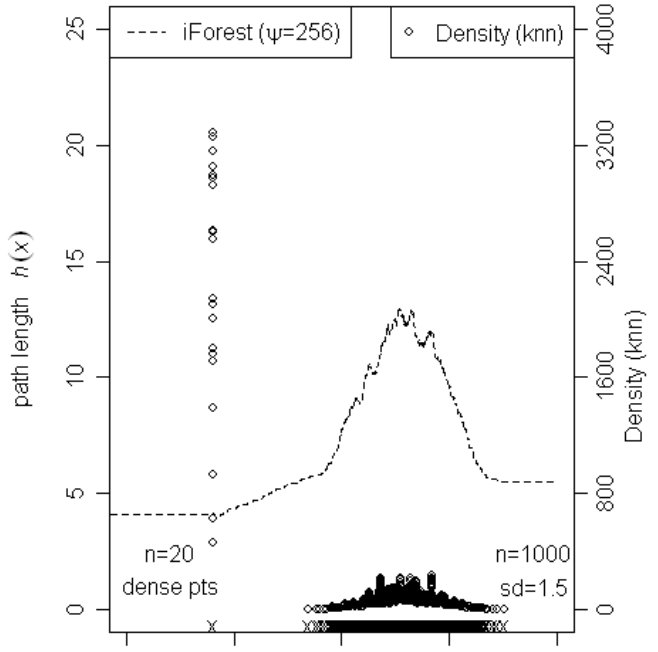
In the following discussion, we argue that the path length as an isolation measure is more appropriate for the task of anomaly detection than the basic density and distance measures.

Using basic density measures, it is assumed that *'normal points occur in dense regions, while anomalies occur in sparse regions'*. Using basic distance measures, the assumption is that *'a normal point is close to its neighbours and an anomaly is far from its neighbours'* (Chandola et al., 2009).

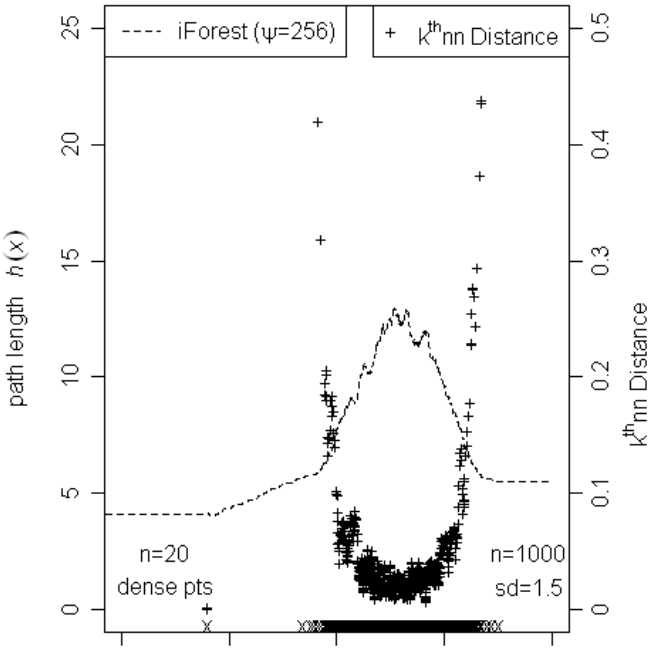
There are situations which violate these assumptions, e.g., a high density and short distance do not always imply normal instances; likewise a low density and high distance do not always imply anomalies. When the density or distance is measured in a local context, which is often the case, points with a high density or short inter-distance can be anomalies in the context of the entire data set. However, there is less ambiguity in the path length-based isolation which we demonstrate in the following three paragraphs.

For a quick recap, in density-based anomaly detection, the anomalies are defined as data points in regions of low density. Density is commonly measured as (a) the reciprocal of the average distance to the k-nearest neighbours and (b) the count of points within a given radius (Tan et al., 2005). In distance-based anomaly detection, the anomalies are defined as data points which are distant from all other points. Two common ways to define distance-based anomaly scores are (i) the distance to the k^{th} nearest neighbour and (ii) the average distance to the k-nearest neighbours (Tan et al., 2005).

On the surface, the function of an isolation measure is similar to a density measure or distance measure, i.e., scattered anomalies are ranked higher than normal points. However, we find that path-length-based isolation behaves differently from a density or distance measure when the anomalies are no longer scattered. In Figure 3.3, at the bottom of the charts, the dense and clustered anomalies on the left are shown together with a large cluster of normal points on the right. The path length, density ($k$nn) and k^{th}nn distance are also plotted. We use $k = 10$ in both the density and distance calculations. In Figure 3.3(a), the density methods show a high density for the anomalies and low density for the normal cluster. In Figure 3.3(b), the k^{th}nn distance reports short distances for the anomalies and larger distances for the points in the normal clusters. According to the anomaly



(a)



(b)

Figure 3.3: This example shows that points with a high density and short inter-distance (the dense points on the left) could well be anomalies. On the other hand, the low density points and points with long inter-distance (the normal distribution) could also be normal instances. Note that the path length is able to correctly detect the dense cluster as anomalies, by giving it a shorter path length. Using a dense cluster of twenty (on the left) and a thousand points (on the right), we compare the path length with the (a) density ($k$nn) and (b) k^{th}nn distance, where $k = 10$.

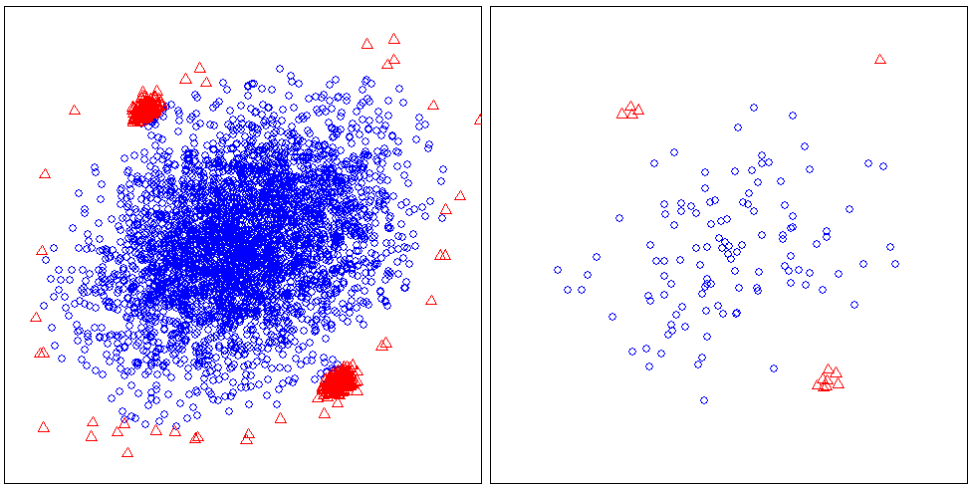
definitions of the density and distance approaches, the anomalies are more "normal" than all the points in the normal cluster. These results are counter-intuitive. The path length, however, is able to address this situation by giving the isolated dense points shorter path lengths. The main reason for this is that the path length is grown in an adaptive context, in which the context of each partitioning is different, from the first partition (the root node) in the context of the entire data set, to the last partition (the leaf node) in the context of the local data-points. Since these dense anomalies are abnormal in the context of the entire data set, many isolation trees would have isolated these dense anomalies early in the partitioning process (global context) and correctly identified them as anomalies. The density ($k$nn) and k^{th}nn distance measures fail because they are only concerned with the k nearest neighbours (local context). This is reflected in the assumptions of the density and distance measures, where dense and clustered anomalies are excluded.

In summary, we have compared three fundamental approaches to detecting anomalies; they are: isolation, density and distance. We have shown an example of detecting dense and clustered anomalies, in which the isolation measure is effective, while the density and distance measures fail. We find that the isolation measure is able to detect both clustered and scattered anomalies; whereas both the distance and density measures can detect scattered anomalies only. This shows that the isolation measure is more appropriate for anomaly detection than the basic density and distance measures. While there are many ways to enhance the basic distance and density measures, e.g., LOF (Breunig et al., 2000), the isolation measure is more elegant because no further 'adjustment' to the basic mechanism is required to detect both clustered and scattered anomalies.

3.2 Better Isolation Models for Masking and Swamping

The problems of swamping and masking have been studied extensively in anomaly detection (Murphy, 1951). Swamping refers to normal instances being wrongly identified as anomalies. For example, when the number of normal instances increases or becomes more scattered. Masking, on the other hand, is the existence of too many anomalies concealing their own presence. It occurs when anomaly clusters become large and dense. An example can be found in Figure 3.3. Under these two circumstances, many anomaly detectors break down.

Isolation-based methods are able to build a model with multiple sub-samples which reduce the influence of swamping and masking. Generally, a larger training sample implies a better detection performance. However, we find that using sub-samples of a small size gives a better performance to the isolation-models than using the entire data set. This is because the sub-samples have fewer normal points 'interfering' with the anomalies; thus, making it easier to isolate the anomalies under swamping and masking.



(a) Original sample (4096 instances)

(b) Sub-sample (128 instances)

Figure 3.4: Using generated data to demonstrate the problems of swamping and masking, (a) shows the original data generated by Mulcross ($n = 4096, a = 0.1, cl = 2, D = 0.8, d = 2$) and (b) shows a sub-sample of the original data. Circles ($\circ$) denote normal instances and triangles ($\triangle$) denote anomalies.

To illustrate this, Figure 3.4(a) shows a data set generated by the Mulcross data generator (Rocke and Woodruff, 1996). We generate two dense anomaly clusters close to a large cluster of normal points. Here the interfering normal points surround the anomaly clusters (swamping effect), and the anomaly clusters are denser than the normal cluster (masking effect). Note that the dense anomaly clusters are still considered as anomalies as they are still less than and different from the normal instances. Figure 3.4(b) shows a training sub-sample of 128 instances of the original data in which the anomaly clusters are clearly identifiable. Those 'spurious' normal instances surrounding the two anomaly clusters cause the swamping effect and have been cleared out. On the other hand, the size of the anomaly clusters causing the masking effect has become smaller. Together, this makes the anomaly clusters easier to isolate, which explains why isolation-based methods are able to utilize small training sub-samples in detecting anomalies.

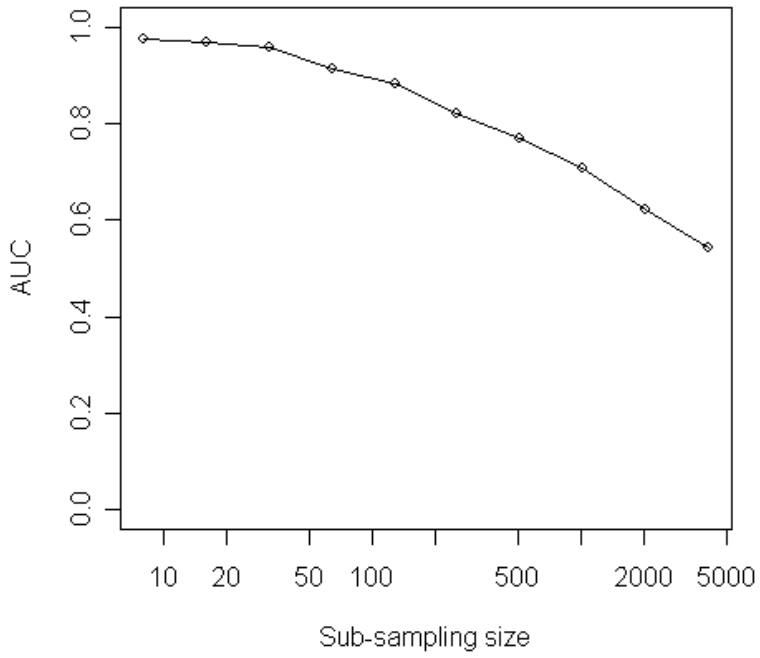


Figure 3.5: Using Mulcross data ($n = 4096, a = 0.1, cl = 2, D = 0.8, d = 2$) as illustrated in Figure 3.4(a), the above diagram shows the detection performance (y-axis) of an isolation-based method (iForest) using various sub-sampling sizes (x-axis in log scale). Due to masking and swamping, the detection performance peaks at a small sub-sampling size.

Figure 3.5 presents the detection performance of this isolation-based method across various sub-sampling sizes. Area Under receiver operating characteristic Curve (AUC) (Fawcett, 2006) is used as the performance measure, the higher the AUC the better the detection performance. AUC of 0.5 means detecting anomalies randomly. When using the entire sample, the isolation-based method reports an AUC of 0.51. When using a sub-sampling size of 128 and 256, it achieves an AUC of 0.91 and 0.83, respectively. The result shows that the isolation-based method is robust to the problems of swamping and masking through significantly reduced training sub-samples.

Regarding the sensitivity of the parameter ψ, an analysis on real life data sets can be found in Section 5.5 of Chapter 5. We also found that when detecting anomalies among arbitrary patterns, a larger ψ is required. More detailed results can be found in Section 5.6 of Chapter 5.

3.3 Chapter Summary

In this chapter, we introduce and illustrate the concept of isolation and we show that anomalies are identifiable with isolation, regardless of whether they are scattered or clustered. Using simple demonstrations, we show that the isolation measure is more suitable for the task of anomaly detection compared to basic density and distance measures. We show that small sub-samples build better isolation models under masking and swamping.

Chapter 4

Algorithmic Framework

In this chapter, we provide an algorithmic framework for isolation-based anomaly detection. This framework serves as a generic template. We will describe several options to this generic framework. This allows the the formulation of different variants through choosing different options in the framework. After that, we will propose two variants, namely iForest and SCiForest which have distinctive abilities in detecting anomalies.

In general, isolation-based anomaly detection is a two-stage process. The initial training stage builds isolation trees using sub-samples of the training set. Then the evaluation stage passes the test instances through isolation trees to obtain an average path length for each instance. This two-stage process is described in Sections 4.1 and 4.2. We will describe a formulation of the anomaly score in Section 4.3, the use of hyper-plane in Section 4.4 and a deterministic split-point selection criterion in Section 4.5. The features of iForest and SCiForest can be found in Section 4.6. Section 4.7 summarizes this chapter.

4.1 Training Stage

In the training stage, an iTree, with the structure of a binary tree, is constructed by recursively partitioning a sub-sample of the given training set until all instances are isolated. An isolation-based model consists of a t number of iTrees. There are two basic input parameters in Algorithm 1 to construct an iTree. They are the sub-sampling size ψ and the number of trees t. The sub-sampling size ψ controls the training sample size available to each iTree. The number of trees t controls the ensemble size of a model. Each iTree is

constructed from an independent sub-sample, randomly sampled from the original training set without replacement. Since iTrees are constructed using different sub-samples, this creates diverse iTrees to form an ensemble model. At the end of the training process, a collection of trees is returned and is ready for path length evaluation.

The algorithms presented in this section are generic and do not specify how the data are being split at each node. In this thesis, we propose two ways to split the data at a node, axis-parallel or non-axis-parallel. In an axis-parallel split, an attribute r is chosen among all the available attributes and a split point p is chosen between the maximum and minimum values of attribute r. A non-axis-parallel split relies upon the use of hyper-planes instead of individual attributes. The use of hyper-planes introduces one more parameter—q the number of randomly selected attributes used in a hyper-plane. The implementation of the hyper-plane will be introduced in Section 4.4.

In the following description, we will describe the tree generation process in generic terms for simplicity. Regardless of the orientation of a split, we propose two options to select a split point, they are: 1) random split-point selection and 2) deterministic split-point selection. In the random split-point selection, a split point is uniformly selected between the maximum and minimum values provided by the split attribute r using the sample at hand. Since, a split point is selected within the maximum and minimum values, it ensures that the resulting nodes have at least one training instance.

The deterministic split-point selection criterion used in this thesis is called $Sd_{reduction}$. It aims to minimize the dispersion of the post-split distributions. Similarly, a split point in deterministic split-point selection is also selected within the minimum and maximum values of the sample at hand. Since, it is possible to examine the criterion over multiple attributes or hyper-planes, another parameter τ is introduced to control the number of attributes or hyper-planes examined at each node. Parameter τ will be introduced in Section 4.5.

Algorithm 1 : $iForest(X, t, \psi)$

Inputs: X - input data, t - number of trees, ψ - sub-sampling size
Output: a set of t *iTrees*

1: **Initialize** $Forest$
2: **for** $i = 1$ to t **do**
3: $\quad X' \leftarrow sample(X, \psi)$
4: $\quad Forest \leftarrow Forest \cup iTree(X')$
5: **end for**
6: **return** $Forest$

Algorithm 2 : $iTree(X)$

Inputs: X - input data
Output: an iTree

1: **if** X cannot be divided **then**
2: return $exNode\{Size \leftarrow |X|\}$
3: **else**
4: select a hyper-plane or an attribute r
5: select a split point p between $\max(r, X)$ and $\min(r, X)$ values of r in X
6: $v \leftarrow \max(r, X) - \min(r, X)$
7: $X_l \leftarrow filter(X, r < p)$
8: $X_r \leftarrow filter(X, r \geq p)$
9: return $inNode\{Left \leftarrow iTree(X_l),$
10: $Right \leftarrow iTree(X_r),$
11: $Size \leftarrow |X|,$
12: $splitPlane \leftarrow r,$
13: $splitValue \leftarrow p,$
14: $upperLimit \leftarrow p + v,$
15: $lowerLimit \leftarrow p - v\}$
16: **end if**

The basic algorithmic framework of the training stage can be found in Algorithms 1 and 2. In Algorithm 1, lines 2 to 5 in the function $iForest$ are a simple for-loop to generate t iTrees by calling the function $iTree$. X' is a sub-sample, of size $|X'| = \psi$, returned by the function *sample* in line 3, randomly sampled from the original training set X without replacement. Line 4 aggregates the individual iTrees into a set of trees, $Forest$, which is then returned at line 6. In Algorithm 1, line 1 of function $iTree$ checks for the condition to stop the tree growing process, which is when the input data X cannot be divided. This condition is met when all members of X are the same or when $|X| < 2$, and an external node $exNode$ is returned. If the stopping condition is not met, line 4 selects an attribute or a hyper-plane r, and line 5 selects a split point p between the maximum and minimum values of r. Lines 7 and 8 divide the data at hand X into X_r and X_l according to r and p. Line 9 continues the tree growing process and returns an internal tree node $inNode$. The members of $inNode$ are: $Left$—a node pointer to the left daughter node, $Right$—a node pointer to the right daughter node, $splitPlane$, $splitValue$, $upperLimit$ and $lowerLimit$. The last two are useful for a facility called the acceptable range, which will be explained in the next section.

4.2 Evaluation Stage

In the evaluation stage, the expected path length $E(h(\mathbf{x}))$ for each test instance $\mathbf{x}$ is calculated by passing the test instance $\mathbf{x}$ through every iTree. Using the *PathLength* function shown in Algorithm 3, a single path length $h(\mathbf{x})$ is derived by counting the number of edges e as instance $\mathbf{x}$ traverses through the iTree from the root node to an external node. In line 1, if the current node T is an external node, then an estimated path length $e + c(T.size)$ is returned at line 2.

iTree	BST
Proper binary tree	Proper binary tree
External node termination	Unsuccessful search
Not applicable	Successful search

Table 4.1: List of equivalent structure and operations in the iTree and Binary Search Tree (BST)

Since iTrees have a structure equivalent to the Binary Search Tree or BST (see Table 4.1), the estimation of the average $h(\mathbf{x})$ for external node terminations is the same as that of the unsuccessful searches in BST. We borrow an analysis from BST to estimate the average path length of iTree. Given a data set of ψ instances, Section 10.3.3 of (Preiss, 1999) gives the average path length of the unsuccessful searches in BST as $c(.)$, which is defined as follow:

$$c(m) = \begin{cases} 2H(m-1) - 2(m-1)/m & \text{for } m > 2, \\ 1 & \text{for } m = 2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

where $H(i)$ is the harmonic number and it can be estimated by $ln(i) + 0.5772156649$ (Euler's constant). The adjustment of the $c(Size)$ in line 2 of Algorithm 3, accounts for estimating the average path length of a random sub-tree, which could have been constructed using $Size$ number of instances beyond the current node. It helps to distinguish between clusters of identical points and isolated single points. Line 4 assigns r as the *splitPlane* of the node. Line 5 sets up i the increment of the node. In line 6, when $\mathbf{x}$ is out of the acceptable range, i.e., $x^r \geq upperLimit$ or $x^r < lowerLimit$, the counting of the path length is omitted, assigning $i \leftarrow 0$. Lines 9 to 13 continue the traversal of $\mathbf{x}$ by calling the *Pathlength* function recursively.

Algorithm 3 : $PathLength(\mathbf{x}, T, e)$

Inputs : **x** - an instance, T - an iTree, e - current path length; to be initialized to zero when first called

Output: path length of **x**

1: **if** T is an external node **then**
2: return $e + c(T.size)$ {$c(.)$ is defined in Equation 4.1}
3: **end if**
4: $r \leftarrow T.splitPlane$, x^r is the value of attribute r in x
5: $i \leftarrow 1$
6: **if** $(x^r \geq T.upperLimit || x^r < T.lowerLimit)$ **then**
7: $i \leftarrow 0$
8: **end if**
9: **if** $x^r < T.splitValue$ **then**
10: return $PathLength(\mathbf{x}, T.left, e + i)$
11: **else**
12: return $PathLength(\mathbf{x}, T.right, e + i)$
13: **end if**

When evaluating the path length on unseen data, setting up an acceptable range at the evaluation stage avoids anomalies that are out-of-range.

An illustration of the acceptable range is shown in Figure 4.1. The effect of the acceptable range can be observed in Figure 4.2. Without the acceptable range, the isolation model forms a loose path length contour in regions that have no training samples as seen in Figure 4.2(a). Loose contour is not able to distinguish between far off and near by anomalies. In situations where there are more than one anomalies in a region of the same path length, those that are closer to the normal points cannot be distinguished from those that are further away. In Figure 4.2(b), the acceptable range helps sharpen the anomaly score contour outside of the normal range. When there are unseen anomalies, the sharpened contour is able to rank them correctly. The anomaly score will be introduced and explained in the next section.

In summary, we have presented the basic framework of isolation-based anomaly detection, which consists of two stages, the training stage and the evaluation stage. In

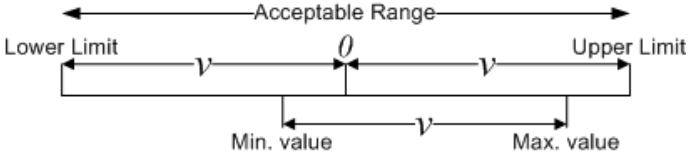
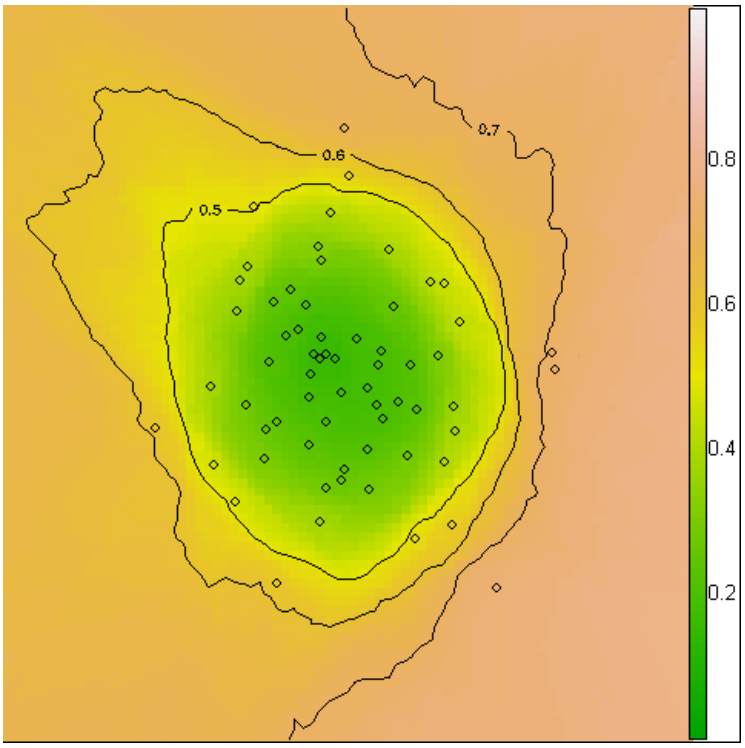
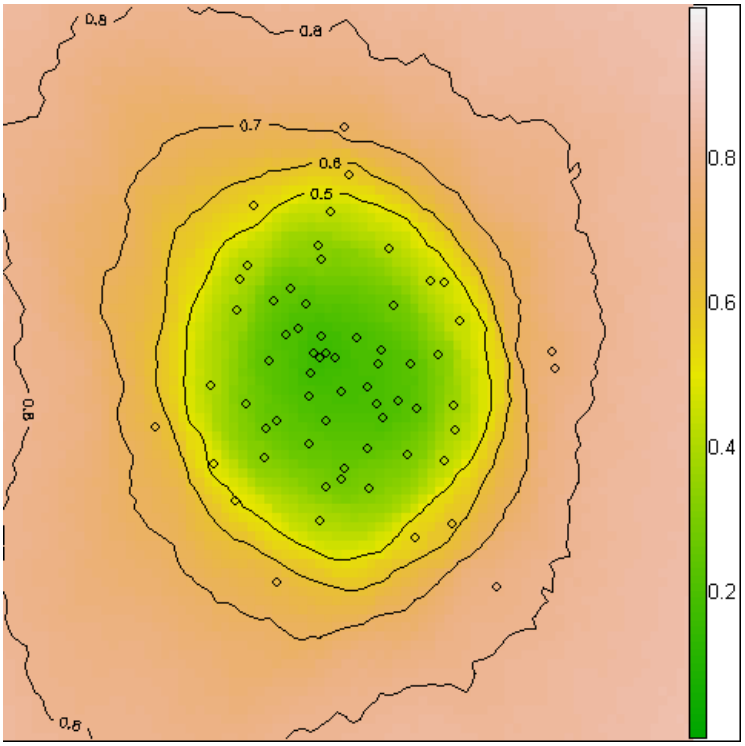


Figure 4.1: Acceptable Range



(a) Without Acceptable Range



(b) With Acceptable Range

Figure 4.2: In (a), without acceptable range, the anomaly score contour outside of the main cluster is very loose. In (b), after the acceptable range is being applied, the outer contour is now sharpened. The sharpened contour helps to rank unseen anomalies properly. These two contours are generated by an isolation-based method that uses non-axis-parallel splits.

the training stage, a set of iTrees is constructed using a given training sample. In the evaluation stage, the path length of every instance is estimated.

4.3 Anomaly Score

An anomaly score is required in any anomaly detection method. The problem of directly using the path length function $h(x)$ as the anomaly score is that path lengths from models of different sub-sampling sizes cannot be directly compared. This is because the maximum possible height of iTree grows in relation to ψ, while the average height grows in relation to $\log \psi$ (Knuth, 1998). Normalizing $h(\mathbf{x})$ by ψ or $log(\psi)$ is either not bounded or cannot be directly compared.

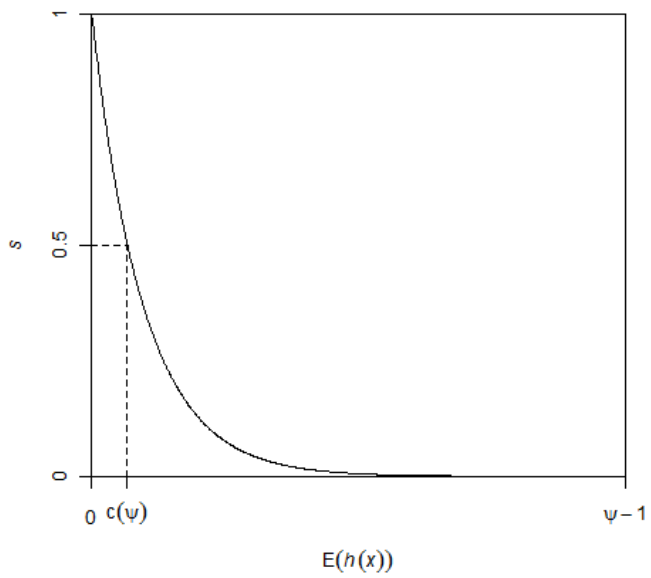


Figure 4.3: The relationship of the expected path length $E(h(\mathbf{x}))$ and the anomaly score s. $c(\psi)$ is the average path length as defined in Equation 4.1. If the expected path length $E(h(\mathbf{x}))$ is equal to the average path length $c(\psi)$, then $s = 0.5$, regardless of the value of ψ.

As $c(\psi)$ (in Equation 4.1) is the average of $h(\mathbf{x})$ given ψ, we can use it to normalise $h(\mathbf{x})$. The anomaly score s of an instance $\mathbf{x}$ is defined as:

$$s(\mathbf{x}, \psi) = 2^{-\frac{E(h(\mathbf{x}))}{c(\psi)}}, \tag{4.2}$$

where $E(h(\mathbf{x}))$ is the average of $h(\mathbf{x})$ from a collection of iTrees. The following conditions provide three special values for this anomaly score:

(a) when $E(h(\mathbf{x})) \to 0$, $s \to 1$;

(b) when $E(h(\mathbf{x})) \to \psi - 1$, $s \to 0$; and

(c) when $E(h(\mathbf{x})) \to c(\psi)$, $s \to 0.5$.

Figure 4.3 illustrates the relationship between $E(h(\mathbf{x}))$ and s, and the range of values are $0 < s \leq 1$ and $0 < h(\mathbf{x}) \leq \psi - 1$. Using the anomaly score s, we are able to make the following assessment:

(i) if the instances return an s very close to 1, then they are definitely anomalies,

(ii) if the instances have an s much smaller than 0.5, then they are regarded as normal instances, and

(iii) if all the instances return $s \approx 0.5$, then the entire sample does not have any distinct anomalies.

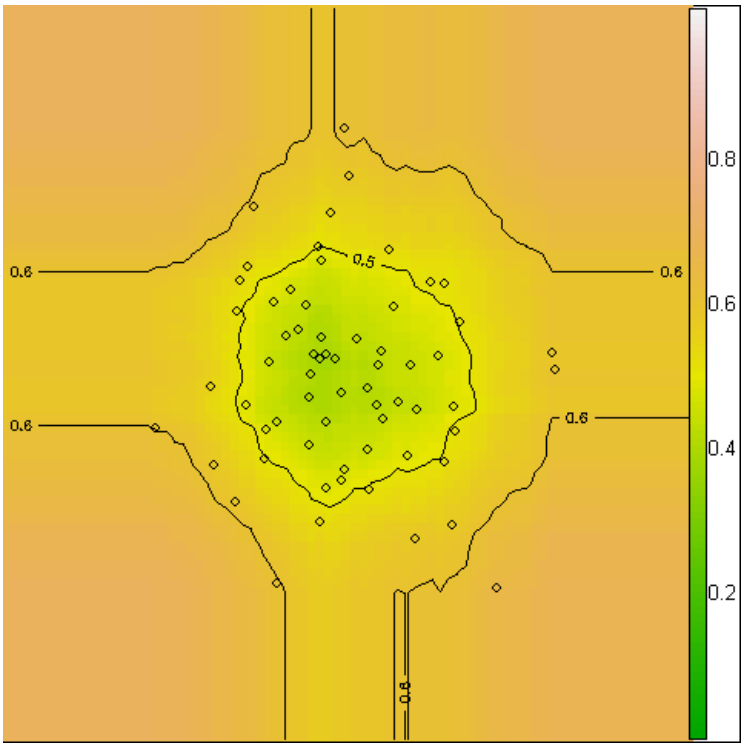
A contour of the anomaly score can be produced by passing a lattice sample through an isolation model, facilitating a detailed analysis of the detection results. Figure 4.4(a) shows an example of such contours in which we can clearly identify three points, for which $s > 0.6$ and they are potentially anomalies.

4.4 Splitting data using hyper-planes

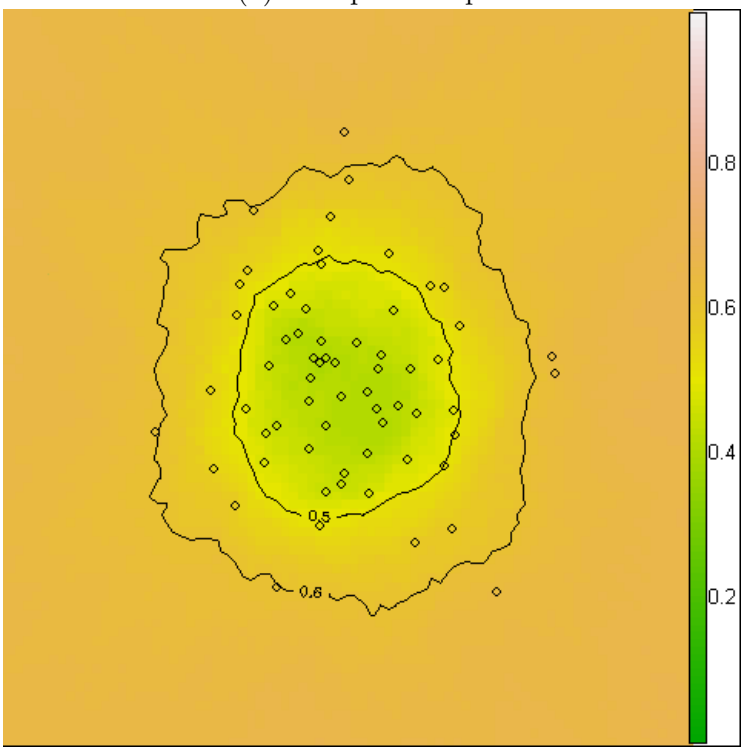
When using axis-parallel splits, we notice that it causes the area with a high value anomaly score, e.g. $s = 0.6$ in Figure 4.4(a) to form a 'cross'. Since the anomaly scores are the same within this cross, there is no way to determine if a point is more abnormal than the others. In order to avoid this problem caused by the axis-parallel splits, a hyper-plane can be used instead of an attribute in selecting a split point. A hyper-plane is defined as:

$$\sum_{j \in Q} \frac{c_j \times x^j}{\sigma_j} = p, c_j \textit{ within the range } \pm(-1, 1], |Q| = q \tag{4.3}$$

The coefficients c_j are randomly generated to create diverse hyper-planes and Q is a subset of q randomly selected attributes. If attributes are not included in Q, then they are not



(a) Axis-parallel split.



(b) Random hyper-planes.

Figure 4.4: Anomaly score contour of two isolation models: (a) using axis-parallel split and (b) using random hyper-planes, for a Gaussian distribution of sixty-four points. Contour lines for $s = 0.5, 0.6, 0.7$ are illustrated. In (a), the axis-parallel split forms a 'cross' on contour line $s = 0.6$ extending from the normal points. In (b), the hyper-planes smooth out the contour lines and avoid the 'cross' effect as compared to (a).

involved in this hyper-plane. The coefficient c_j is randomly generated within the range of $\pm(-1, 1]$ and p is a split point selected by a split selection criterion.

As each attribute has a different range, normalization is required to ensure the attributes are treated equally. Each attribute is normalized by the standard deviation of the sub-sample attribute values σ_j taken at the root node of each tree.

In a hyper-plane, the absolute value of the coefficient c_j represents the involvement of the attribute j. Setting c_j to 0 turns off the involvement of attribute j.

Using randomly generated hyper-planes, the 'cross' is removed. Figure 4.4(b) shows the effect of randomly generated hyper-planes using random split-point selection. Note that the outer contour line $s = 0.6$ now resembles the shape of the Gaussian distribution and the line is able to determine the outlying points.

In terms of implementing a hyper-plane to the isolation-based framework, an additional parameter q at the training stage is required to determine how many randomly selected attributes are involved in forming the hyper-planes. We find that the detection performance is sensitive to q depending on the nature of the anomalies. In Chapter 5, Section 5.7 will expound on the relationship between q and the detection performance.

4.5 Selecting Split Based on a Reduction of Standard Deviation

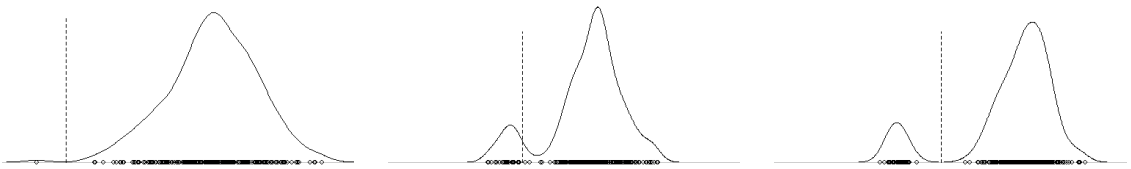
At each internal node of an iTree, there are two alternative ways to split the data. The first alternative is to select the split point randomly between the maximum and minimum values of a randomly selected attribute or hyper-plane. This is a quick and simple way that requires minimal effort. The second alternative is to deterministically find the best split point according to the data distribution. It is intuitive to separate the normal instances from anomalies at every node. However, since the learning is unsupervised, we can only assume that the data points that are very different from the rest are anomalies.

This split-selection criterion is partly motivated by Hawkin's outlier definition (Hawkins, 1980), *"An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism."* Thus, based on the different data distributions of anomalies, we are able to determine a good separation point to clearly separate the anomalies from normal instances.

When a split clearly separates two different distributions, their dispersions are minimized. Using this simple mechanism, our proposed split-selection criterion ($Sd_{reduction}$) is defined as:

$$Sd_{reduction}(p, Y) = \frac{\sigma(Y) - avg(\sigma(Y^l), \sigma(Y^r))}{\sigma(Y)}, p \text{ separates } Y \text{ into } Y^l, Y^r, \tag{4.4}$$

where $Y^l \cup Y^r = Y$; Y is a set of values of q. $\sigma(.)$ is the standard deviation function and $avg(a, b)$ simply returns $\frac{a+b}{2}$. p is the split point that separates Y into Y^l and Y^r. The criterion is normalised using $\sigma(Y)$, which allows the comparison of different scales between different attributes. To find the best split for a given sample Y, we pass the data twice. The first pass computes the base standard deviation $\sigma(Y)$ while the second pass finds the best split p which gives the maximum $Sd_{reduction}$ using Equation 4.4. When an anomaly cluster is present in Y, it is first separated as this is the most effective way to reduce the average standard deviation of Y^l and Y^r. To calculate the standard deviation, a reliable one-pass solution can be found in (Knuth, 1968, p. 232, vol. 2, 3rd ed.). This solution is not subject to cancellation error[1] and keeps the computational cost to a minimum.



(a) Separating an anomaly from the main distribution

(b) Dividing a group of local clustered anomalies close to the main distribution

(c) Isolating a group of clustered anomalies from the main distribution

Figure 4.5: Examples of the $Sd_{reduction}$ split point in three univariate distributions.

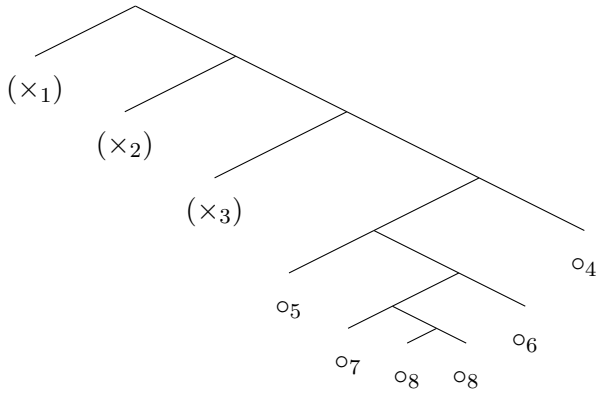
We illustrate the effectiveness of $Sd_{reduction}$ in Figure 4.5. This criterion is shown to be able to (a) separate a normal distribution from an anomaly, (b) isolate a cluster of anomalies which are very close to the main distribution, and (c) separate an anomaly cluster from the main distribution. With the help of the analysis in (Schilling et al., 2002), we find that $Sd_{reduction}$ is able to separate the two distributions early in the tree construction process, as long as the combined distribution of any two normal distributions is bimodal. In order to assess the effectiveness of $Sd_{reduction}$ in separating the data of

[1]The cancellation error refers to the inaccuracy in computing very large or very small numbers, which are out of the precision of ordinary computational representation.

different distributions, we first consider two distributions with the same variance i.e., $\sigma_1^2 = \sigma_2^2$, and means μ_1 and μ_2. It has been shown that the combined distribution can only be bimodal when $|\mu_2 - \mu_1| > 2\sigma$ (Schilling et al., 2002). In the case where $\sigma_1^2 \neq \sigma_2^2$, the condition of bi-modality is $|\mu_2 - \mu_1| > S(r)(\sigma_1 + \sigma_2)$, where the ratio $r = \sigma_1^2/\sigma_2^2$ and separation factor $S(r) = \frac{\sqrt{-2+3r+3r^2-2r^3+2(1-r+r^2)^{\frac{3}{2}}}}{\sqrt{r}(1+\sqrt{r})}$ (Schilling et al., 2002). S equals 1 when $r = 1$ (i.e. $\sigma_1^2 = \sigma_2^2$), and S decreases slowly when r increases. This means that the bi-modality holds when the one-standard deviation regions of the two distributions do not overlap. This condition is generalised for any ratio between the two distributions and is further relaxed when the standard derivations are different. From this condition of bi-modality, it is clear that $Sd_{reduction}$ is able to separate any two distributions as long as their one-standard deviation regions do not overlap.

In Figure 4.5(c), the behaviour of setting a split point between the large and the smaller clusters is actually treating isolated points around these clusters as anomalies. Further splits will isolate these points from these clusters. In Figure 4.5(b), the behaviour to set a split point inside the smaller cluster is more in fact desirable than separating the smaller cluster from the main cluster. Because the path lengths of all the clustered anomalies ($\times$) will be shorter than the shortest path length of a normal point ($\circ$). To illustrate this, we present an example tree below:

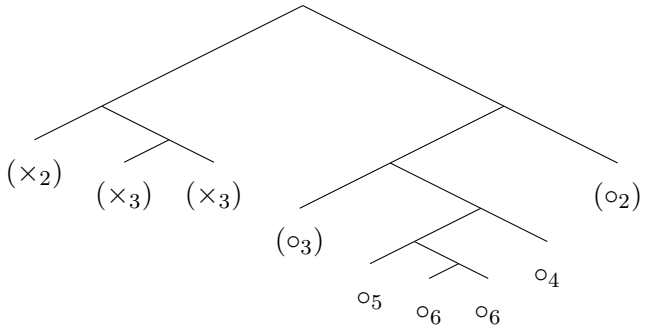
Tree resulting from isolating clustered anomalies first



The reason for this behaviour is a tendency of $Sd_{reduction}$ to delay splitting larger clusters. This is because the standard deviation of larger cluster remains mostly the same when a few outlying points are included. However, the standard deviation of a smaller cluster increases dramatically when those points are included.

In contrast, when an anomaly cluster is separated from the normal cluster at the first split, the anomalies are easily confused with normal instances with short path lengths as demonstrated in the following example:

Tree resulting from separating anomaly cluster from normal cluster first



Besides standard deviation, we also experiment with the mean absolute deviation and variance in similar formulations. We find that only the standard deviation is able to select split points which are beneficial for detecting anomalies as shown in Figure 4.5.

4.6 The Two Variants: iForest and SCiForest

In this chapter, we have introduced the hyper-plane split and the axis-parallel split in Section 4.4. We also have introduced the deterministic split-point selection and the random split-point selection in Section 4.5. Rather than enumerating all possible variants, we focus on two variants which have distinctive characteristics which will be elaborated on in Chapter 5.

In this thesis, iForest utilizes axis-parallel splits with random attribute selection and random split-point selection. iForest stands for **I**solation **Forest**. These settings allow iForest to be very efficient without the need to compute hyper-planes or split selection criteria. It is used as the basic model for use in general situations.

The second variant, SCiForest, is specially designed to detect clustered anomalies. SCiForest stands for **I**solation **Forest** with **S**election **C**riterion. SCiForest utilizes hyper-planes with deterministic split-point point selection. We find that the best performance in detecting local clustered anomalies is found using this variant.

Switching from one variant to the other only requires changing lines 4 and 5 in Algorithm 1 in Section 4.2. The implementation of SCiForest requires lines 4 and 5 to be replaced with a for-loop that repeats τ times. Inside the for-loop, we first construct a

	iForest	**SCiForest**
Split	axis-parallel	hyper-plane
Selection criterion	random	$Sd_{reduction}$
Targeting	scattered anomalies	clustered anomalies
Speed	fast	slightly slower

Table 4.2: A quick comparison table for the two variants: iForest versus SCiForest.

	iForest	**SCiForest**
Sub-sampling size ψ	256	256
Number of trees t	100	100
Number of attributes q used in hyper-planes	NA	2
Number of trials τ for $Sd_{reduction}$'s evaluation	NA	10

Table 4.3: Default parameter settings of isolation-based methods.

random hyper-plane. Then, the hyper-plane values are sorted. Lastly, an $Sd_{reduction}$ criterion is applied to find the best split point. The hyper-plane that yields the maximum $Sd_{reduction}$'s value is used.

Other variants have been experimented upon, however, their results are not as interesting and thus are not included in this thesis. Both iForest and SCiForest utilize the acceptable-range since there is no performance loss in using this facility. A quick comparison can be found in Table 4.2. Section 5.4 in Chapter 5 provides illustrative examples in terms of the type of anomalies they are targeting. Section 6.3 in Chapter 6 empirically examines the efficiency of both methods in terms of their processing speed.

The default settings used throughout this thesis for iForest and SCiForest can be found in Table 4.3. The sensitivities of parameters t and ψ are analysed in Section 5.5. Parameters q and τ are examined in Sections 5.7 and 5.8 of Chapter 5. Parameter q controls the number of randomly selected attributes involved in the hyper-planes and is useful for identifying anomalies that depend on multiple attributes. Parameter τ controls the number of trials in the $Sd_{reduction}$ evaluation which searches for the best hyper-plane and split point to isolate any clustered anomalies. It becomes insensitive after a certain value, e.g., $\tau > 10$ as evident in Section 5.8. Split selection trials are useful in improving the detection performance for high dimensional data as shown in Section 7.3.1 of Chapter 7.

The time complexities for the training and evaluating in iForest are $O(t\psi^2)$ and $O(nt\psi)$ respectively. In the training stage, t trees are constructed with maximum height of $\psi - 1$.

For each node, the most time consuming task is to find the maximum and minimum values and divide the data according to a split point, which costs ψ. At the evaluation stage, an n number of instances traverse t trees with a maximum traversal of $\psi - 1$, which is the maximum height of an iTree.

The time complexities of SCiForest are $O(t\tau\psi(q\psi + \psi \log \psi + \psi))$ and $O(qnt\psi)$ respectively for the training and evaluating stages. At the training stage, SCiForest is only different from iForest in constructing the nodes. The three major components in constructing a node are: i) calculating the hyper-plane values, ii) sorting the sample and iii) computing the $Sd_{reduction}$ criterion. Their complexities are $\tau q\psi$, $\tau\psi \log \psi$ and $\tau\psi$, respectively. In the evaluation stage, additional time is taken to calculate the hyper-plane values at each node, which is in the order of q.

Note that, in both methods, the training complexities are constant and only the evaluation complexities grow linearly with n. Since both methods require storage for only t binary trees, the space complexity for both methods is $O(t\psi)$.

4.7 Chapter Summary

This chapter introduces the algorithmic framework of isolation-based anomaly detection. We have described in details the two stages of this framework. We also introduce the two options available in the isolation framework; they are the hyper-plane split versus axis-parallel split, and the deterministic split-point selection versus random split-point selection. Between these two options, we formulate two distinctive variants, called iForest and SCiForest.

Chapter 5

Characteristics of iForest and SCiForest

This chapter is devoted to examining the different characteristics of iForest and SCiForest proposed in Chapter 4. The objectives of this chapter are to:

- highlight that SCiForest is able to handle a difficult type of anomalies known as local clustered anomalies.
- examine the conditions under which iForest and SCiForest will breakdown.
- understand the behavioural differences between iForest and SCiForest.
- examine their ability to detect anomalies using classical statistical data sets.
- examine the effects of the different parameter settings in iForest and SCiForest.
- illustrate ways to visualize anomalies using iForest's and SCiForest's model.
- investigate iForest's and SCiForest's abilities to detect anomalies among arbitrary patterns.

For comparison, three state-of-the-art anomaly detectors are investigated in this chapter, they are ORCA (Bay and Schwabacher, 2003), LOF (Knorr and Ng, 1998) and SVM (Schölkopf et al., 2000). They are selected to represent the best distance-, density- and model-based methods. Details of their implementations can be found in Appendix A.4.

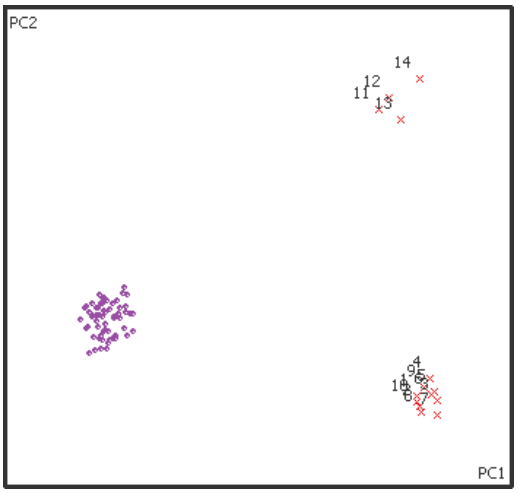
The organization of this chapter is as follows: In Section 5.1, we demonstrate iForest's and SCiForest's abilities to detect anomalies using three statistical data sets. In Section

5.2, we show that SCiForest is able to detect a very difficult class of anomalies, namely local clustered anomalies. In Section 5.3, two analyses are conducted to examine the breakdown properties of SCiForest and iForest. The first analysis examines the breakdown property in terms of an increasing number of anomalies. The second analysis examines the breakdown property in terms of how close the anomalies can be to the normal instances. In Section 5.4, we further examine the behavioural difference between SCiForest and iForest by characterizing the anomalies they identify. In Section 5.5, we examine the sensitivity of the parameters t and ψ that are common to iForest and SCiForest. In Section 5.6, using four synthetic data sets with arbitrary patterns, we demonstrate that iForest and SCiForest are able to detect anomalies among arbitrary patterns. We also find that isolation-based methods require a higher sub-sampling size in detecting anomalies among arbitrary patterns. In Sections 5.7 and 5.8, we explore how the parameters q and τ in SCiForest can be adjusted to improve the detection accuracy. In Section 5.9, we look at how we can use iForest and SCiForest to visualize anomalies. In Section 5.10, we summarize the common and different characteristics of these two methods.

5.1 Statistical data sets

In this section, we use three statistical data sets: *hbk*, *starsCYG* and *wood* (Rousseeuw and Leroy, 1987) to demonstrate iForest's and SCiForest's abilities in detecting anomalies. The *hbk* data set was artificially generated by Hawkins et al. (1984). The *hbk* data set consists of 75 instances of 4 attributes. It is a good demonstration of the masking effect since all 14 anomalies are grouped into two clusters. The *starsCYG* data set (Rousseeuw and Leroy, 1987) is created from the Hertzsprung-Russell Diagram of Star Cluster CYG OB1, which contains 47 stars. Two variables are the logarithm of the effective temperature at the surface of the star and the logarithm of its light intensity. The *wood* data set originated from (Drapper and Smith, 1966), where it was used to study the relationship between anatomical factors and wood specific gravity. The *wood* data set has 20 instances in 6 attributes. The *wood* data set was contaminated by replacing 4 instances with clustered anomalies (Drapper and Smith, 1966).

The above statistical data sets are relatively small, however they are not toy problems. They are designed to examine the different aspects of anomaly detectors. In our comparison, we use these data sets to illustrate three scenarios which are difficult for different anomaly detectors.

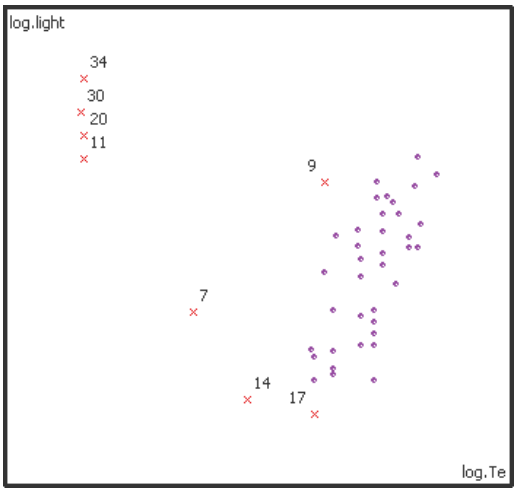


rank	iF	SCiF	ORCA	SVM	LOF
1	**14**	**14**	**14**	**1**	**14**
2	**12**	**13**	**12**	**3**	47
3	**13**	**7**	**11**	**4**	68
4	**11**	**12**	**13**	**7**	16
5	**10**	**11**	**7**	**10**	61
6	**4**	**1**	47	**11**	34
7	**7**	**10**	**4**	**12**	53
8	**1**	**6**	**3**	**13**	**12**
9	**2**	**2**	**1**	**14**	52
10	**8**	**3**	**10**	16	42
11	**9**	**8**	**6**	43	37
12	**3**	**5**	70	44	75
13	**5**	**4**	53	47	44
14	**6**	**9**	25	49	36
15	47	52	68	60	31
16	52	68	43	68	70
17	68	47	**8**	**2**	62
18	53	53	**2**	**5**	30
19	43	16	62	**6**	35
20	60	15	**5**	**8**	43

Figure 5.1: Evaluation of the *hbk* data set (Scenario: anomalies with varying densities). (left) Visualization of the *hbk* data with its first two principle components, datum 1 to 10 are located at the bottom right corner as a dense cluster, datum 11 to 14 are located at the top right corner as a scattered cluster. (right) Evaluation ranking of *hbk* data set, bold faced datum indexes are actual anomalies.

Our first statistical data set is the *hbk* data set. Here the anomalies comprise of two small clusters with one more scattered than the other, which showcases the scenario of anomalies with varying densities. This condition causes some distance-based and density-based methods to fail, due to the varying anomaly density. Figure 5.1 features a scatter plot of the first two principle components[1] established from the original 4 attributes. It also shows the top 20 anomalies identified by the five anomaly detectors. The result shows that iForest and SCiForest are the only detectors that can correctly rank all the anomalies at the top of the list.

[1]Principle components are used for visualization purpose only; the detection results are obtained using the original attributes.

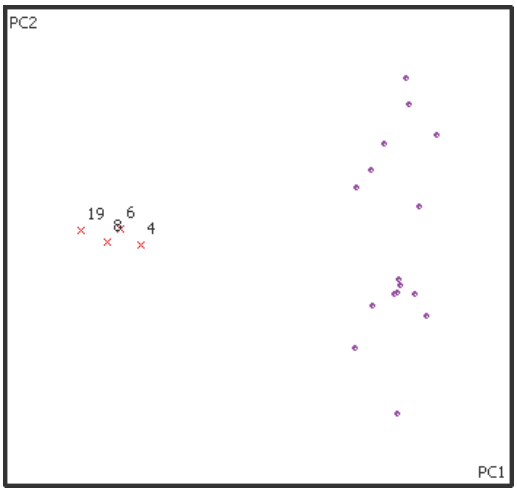


rank	iF	SCiF	ORCA	SVM	LOF
1	**34**	**34**	**34**	2	**11**
2	**30**	**7**	**30**	3	**34**
3	**20**	**11**	**20**	4	**30**
4	**11**	**30**	**11**	**7**	**20**
5	**7**	**20**	**7**	**9**	**7**
6	**14**	**14**	**14**	14	**14**
7	**17**	**9**	**9**	**17**	**17**
8	36	**17**	**17**	18	2
9	2	18	18	25	4
10	4	3	3	29	**9**
11	18	36	36	**30**	36
12	**9**	27	27	32	40
13	19	19	2	44	3
14	8	2	4	47	45
15	3	4	5	1	24
16	29	29	24	5	28
17	45	35	28	6	19
18	22	21	45	8	25
19	35	5	40	10	46
20	32	22	19	**11**	6

Figure 5.2: Evaluation of *starsCYG* (Scenario: presence of both clustered and scattered anomalies). (left) Visualization of the *starsCYG* data with its two attributes, datum 11, 20, 30, 34, 9, 7, 14 and 17 are located at the upper left corner and scattered around the main cluster. (right) Evaluation ranking of *starsCYG* data set, bold faced datum indexes are actual anomalies.

Our second statistical data set is the *starsCYG* data set. The anomalies here are presented as both a small cluster and scattered points around the main cluster. Figure 5.2 features a scatter plot of the two original attributes and shows the top 20 anomalies identified by the five detectors. It is observed that SVM concentrates on detecting the scattered anomalies around the main cluster and neglects the small anomaly cluster. SCiForest and ORCA are the only detectors that correctly rank all the anomalies at the top of the list. Both iForest and LOF misrank one anomaly.

Our third statistical data set is the *wood* data set, it consists of 20 instances of which 4 are anomalies. All the anomalies are located in a dense cluster. Figure 5.3 features a scatter plot of the first two principle components established from the original 6 attributes. It also shows the top 10 instances identified by the five anomaly detectors. The irregular density of the normal points is difficult for SVM, since excluding some of the normal points in a sparse area may result in a smaller enclosing hyper-sphere. These normal points are treated as anomalies, resulting in a poor performance for SVM. SCiForest, ORCA and LOF identify all four anomalies correctly, while iForest misranks one instance only.



rank	iF	SCiF	ORCA	SVM	LOF
1	**19**	**4**	**19**	10	**19**
2	**8**	**19**	**8**	11	**8**
3	10	**8**	**6**	12	**6**
4	**4**	**6**	**4**	**19**	**4**
5	**6**	11	10	1	7
6	20	10	7	2	10
7	12	12	12	3	12
8	7	7	11	**4**	18
9	11	9	9	5	13
10	13	13	13	**6**	11

Figure 5.3: Evaluation of the *wood* data set (Scenario: normal points of varying densities with clustered anomalies). (left) Visualization of the wood data with its two principle components, datum 4, 6, 8 and 19 are located on the right hand side as a dense cluster. (right) Evaluation ranking of wood data set, bold faced datum indexes are actual anomalies.

In this evaluation, we have illustrated three different scenarios: a) anomalies of varying densities, b) the presence of both clustered and scattered anomalies and (c) normal points of varying densities with clustered anomalies. We find that iForest and SCiForest have the least errors across all these scenarios. It shows that iForest and SCiForest are better at handling these different situations.

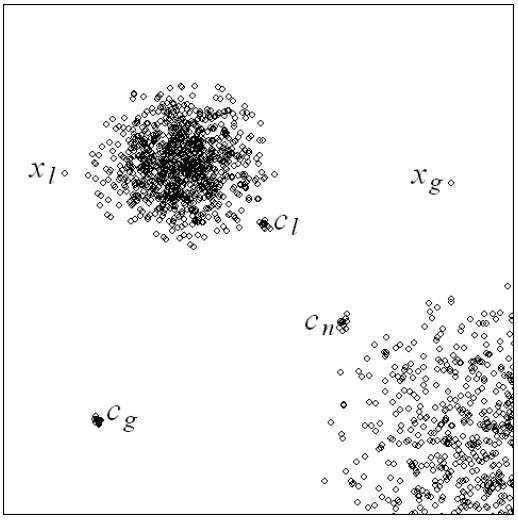
5.2 SCiForest is able to detect local clustered anomalies

In this section, we aim to show that SCiForest is able to cope with the very difficult problem of local clustered anomalies. Local clustered anomalies are difficult because they are located very close to the normal points and are denser than the normal points, which undermine the assumptions of conventional distance- and density-based methods.

In this section, we first divide the anomalies into four types according to their dispersion and locality. Then, using these four types of anomalies, we illustrate a scenario showing SCiForest's ability to detect the local clustered anomalies. Finally, we discuss why local clustered anomalies are challenging for other detectors.

The terms "local", "global", "scattered" and "clustered" are useful for describing different types of anomalies and are defined in Chapter 2. Using combinations of these terms, we are able to describe four types of anomalies: global scattered anomalies, local scattered anomalies, global clustered anomalies and local clustered anomalies. x denotes scattered

anomalies; c denotes clustered anomalies; subscript g denotes global anomalies, and l, n local anomalies. We illustrate these four types of anomalies in a scenario in Figure 5.4(a). Here, each anomaly cluster has twelve data points and the two normal clusters have different densities.



(a) **Data Distribution**

	x_g	x_l	c_g	c_n, c_l
SCiForest	**7**	**38**	**1-6,8-13**	**14-37**
iForest	**12**	28	**1-11,13**	-
LOF($k = 15$)	**1**	**14**	**2-13**	-
LOF($k = 10$)	**1**	**2**	-	-
ORCA($k = 15$)	**1**	27	**2-13**	-
ORCA($k = 10$)	**1**	10	-	-

(b) **Rankings of Anomalies**
Local clustered anomalies c_n, c_l are difficult to detect. '-' means the ranking is > 38. Consecutive rankings are bold-faced. Non-consecutive rankings indicate that false-positives are ranked higher than anomalies.

Figure 5.4: SCiForest is the only detector that is able to detect local clustered anomalies c_n and c_l including all other anomalies in the data set above. (a) illustrates the data distribution and (b) reports the anomaly rankings provided by four different anomaly detectors.

The ranking results of LOF, ORCA, iForest and SCiForest are provided in Figure 5.4(b). In this evaluation, SVM is omitted because fine tuning is required for SVM to handle clustered anomalies. There are 38 anomalies in total and all the methods are able to detect the global scattered anomalies x_g and local scattered anomalies x_l. As for the global clustered anomalies, iForest is able to detect c_g while LOF and ORCA require a larger k for successful detection. Among all the methods, only SCiForest is able to detect the local clustered anomalies c_n and c_l, and correctly rank all anomalies at the top of the list. The difficulties of detecting clustered anomalies can be summarized in the following three points:

- *Large number of anomalies* — when the **plurality** of the clustered anomalies is more than a certain threshold, e.g., the k parameter of the k-nn based methods, then the clustered anomalies become undetectable with $k$nn based methods;

- *High density of clustered anomalies* — when the **density** of the anomalies increases, common measures, such as density and distance, mistake these anomalies as more 'normal' than normal instances;

- *Local anomalies* — when anomalies are located in a close **proximity** to the normal instances, they are easily mistaken as normal instances.

In response to the first two points, both iForest and SCiForest are able to detect clustered anomalies, as long as the ratio of the anomaly population is smaller than that of the normal points and they are detectable by a path length measure regardless of their actual population and density. For the last point, SCiForest has the advantage in separating the data from distinct distributions using $Sd_{reduction}$ which is why SCiForest is able to detect the local clustered anomalies better than iForest. The major difference between the performance of SCiForest and iForest is that SCiForest is able to detect the local clustered anomalies. However, this ability comes with a small increase in the processing time, which will be discussed in Section 6.3 of Chapter 6.

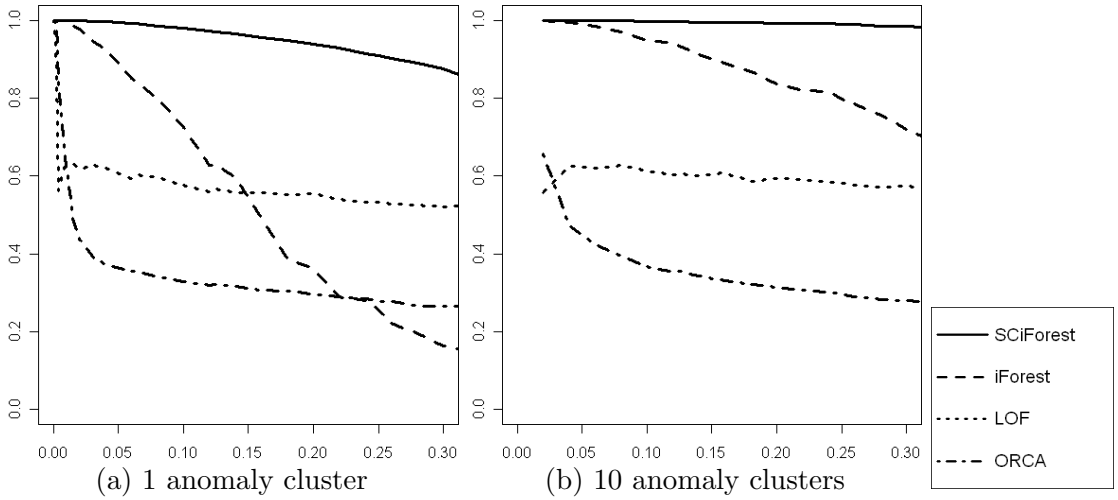
5.3 Breakdown Analyses

This section features two breakdown analyses of the two properties of anomalies, 'few' and 'different'. Intuitively, it is easy to detect anomalies that are small in number and far from the normal points. However, what if these properties change? The following two analyses provide insights into how these properties affect the detection performance of both iForest and SCiForest compared to ORCA and LOF. The results for SVM are omitted in these analyses to keep the diagram readable. In these two analyses, we consistently find that SCiForest has the best breakdown properties. We will discuss why this is the case at the end of this section.

5.3.1 Breakdown property with increasing number of anomalies

In this subsection, we examine the breakdown characteristics of iForest and SCiForest in detecting size-increasing clustered anomalies. For comparison, we also examine the breakdown characteristics of ORCA and LOF.

In order to examine the detection accuracy (AUC) of the different detectors, we use the Mulcross data generator (Rocke and Woodruff, 1996) to generate 4096 normal instances



(a) 1 anomaly cluster (b) 10 anomaly clusters

Figure 5.5: Detection performance of four anomaly detectors on the Mulcross data set with increasing numbers of anomalies. The y-axis presents the AUC performance of SCiForest, iForest, ORCA and LOF. Mulcross setting used is $n = 4096 * (1 + a), cl = \{1, 4\}, D = 1, d = 2$, under various contamination levels in x-axis $a = \{0.02, ..., 0.48, 0.5\}$, with 1 and 10 anomaly clusters in diagrams (a) and (b) respectively.

with anomalies of different contamination levels (percentage of anomalies in the data) and different numbers of anomaly clusters. Information about Mulcross data generator can be found in Appendix A.2. AUC generally drops as the contamination level increases; a sharp drop in AUC indicates an anomaly detector breaks down at a particular contamination level. We examine the range of contamination levels between $a = 0.02$ and $a = 0.3$.

In Figure 5.5, iForest, LOF and ORCA have significantly lower detection performance throughout the different contamination levels compared to SCiForest. Note that, dense clustered anomalies mislead other anomaly detectors more than SCiForest—most of them break down before the contamination level even reaches $a = 0.1$; SCiForest is the most robust detector and breaks down more gradually as the contamination level increases in the case of one anomaly cluster. Note that the scenario with ten anomaly clusters is easier than the one with one anomaly cluster; this is because at the same contamination level, the same number of anomalies are now grouped into ten clusters rather than one big cluster. SCiForest almost maintains its detection performance for the entire range of different contamination levels. However, the scenario with ten anomaly clusters only has a marginal effect on LOF and ORCA; the high volume and density of clustered anomalies still pose difficulties.

5.3.2 Breakdown property with local anomalies

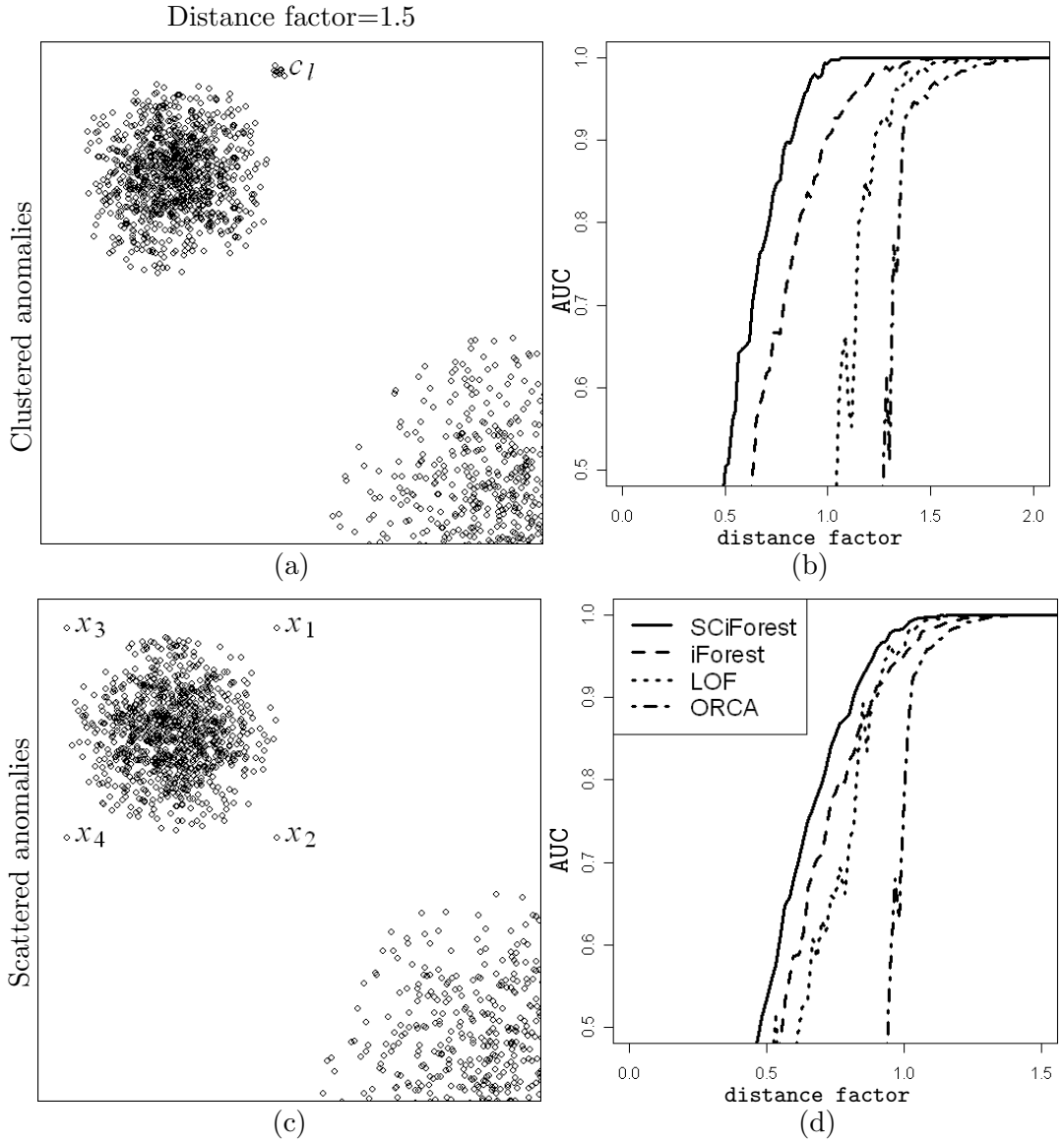


Figure 5.6: Performance in detecting Local Anomalies. (a) and (c) illustrate the data distributions with clustered and scattered anomalies with distance factor = 1.5. The results are shown in (b) and (d) with AUC (y-axis) versus distance factor (x-axis).

When clustered anomalies come close to the normal instances, the anomaly detectors based on density and distance measures break down due to the proximity of the anomalies. To examine the robustness of different detectors against local clustered anomalies, we generate a cluster of twelve anomalies at various distances from either of two normal clusters of different densities. We use distance factor $\frac{D}{r}$, where D is the distance between the clustered anomalies and the centre of a normal cluster and r is the radius of the top-left normal cluster. When the distance factor is equal to 1, the clustered anomalies are located right at the edge of the dense normal cluster. In this evaluation, LOF and ORCA

are given $k = 15$, which is larger than the size of an anomaly cluster, so that they are able to detect these anomalies.

Figure 5.6(a) illustrates the scenario of the clustered anomalies having a distance factor equal to 1.5. As shown in Figure 5.6(b), the results confirm that SCiForest has the best performance in detecting the local clustered anomalies, followed by iForest, LOF and ORCA. When the distance factor is equal to or slightly less than 1, SCiForest's AUC remains high despite the fact that the local anomalies have come into contact with the normal instances.

A similar evaluation is also conducted for scattered anomalies. Figure 5.6(c) illustrates the scenario where the distance factor is equal to 1.5. In Figure 5.6(d), SCiForest also has the best performance in detecting the local scattered anomalies, followed by LOF, iForest and ORCA. Note that LOF is slightly better than iForest from a distance factor of > 0.7 onwards.

SCiForest's better breakdown properties are attributed to the use of the $Sd_{reduction}$ split selection criterion and hyper-planes. In the scenario in Section 5.3.1, $Sd_{reduction}$ causes SCiForest to first separate the clustered anomalies from their nearby normal cluster, then the hyper-planes allow SCiForest to perform non-axis-parallel splits, which maximizes this separation. When the normal points and anomalies are separated early in the tree structure, the path length measure can better represent the difference between normal points and anomalies and hence achieves a better detection performance. For the scenarios presented in Section 5.3.2, the local anomalies can be detected by isolating them from their immediate context. In SCiForest, $Sd_{reduction}$ has a tendency to separate the dense outlying clustered anomalies first and then the sparse fringe normal points before separating the dense points inside a cluster of normal instances. Because of this, SCiForest's partitions are concentrated around the fringe points of a normal cluster, which provide distinct path lengths for the fringe points as compared to the normal points in the centre of a normal cluster. This explains why SCiForest is able to detect the local anomalies even when they are on the fringe of the normal instances.

(a) Data set: *Annthyroid*

id	tsh	t3	tt4	t4u	tfi	tbg	id	tsh	t3	tt4	t4u	tfi	tbg
SCiForest							iForest						
*3287	-1.7	**21.5**	-2.0	-2.9	1.1	-3.0	1645	-1.5	-0.2	**21.2**	**8.9**	-1.6	**14.6**
*5638	-1.8	**20.6**	-1.4	-1.8	1.7	-2.2	2114	1.3	-0.2	**15.0**	**8.4**	-1.0	11.2
*1640	1.5	**21.3**	-2.0	-2.7	2.2	-2.9	*3287	-1.7	**21.5**	-2.0	-2.9	1.1	-3.0
*2602	-1.4	**19.8**	-2.0	-2.4	2.1	-2.7	*1640	1.5	**21.3**	-2.0	-2.7	2.2	-2.9
*4953	-2.6	**20.3**	-0.4	-2.1	1.0	-2.3	3323	1.7	0.4	**6.2**	**4.7**	-0.7	**6.0**
*5311	-1.4	**20.2**	-1.7	-2.5	0.6	-2.6	*6203	-1.8	**18.9**	-2.0	-2.4	1.8	-2.6
*5932	0.4	**22.9**	0.0	-2.8	0.7	-2.9	*2602	-1.4	**19.8**	-2.0	-2.4	2.1	-2.7
*6203	-1.8	**18.9**	-2.0	-2.4	1.8	-2.6	2744	-1.2	0.4	**4.8**	**4.7**	-1.0	6.7
*1353	0.1	**18.8**	-1.4	-2.7	0.2	-2.8	*4953	-2.6	**20.3**	-0.4	-2.1	**1.0**	-2.3
*6360	0.4	**17.2**	-2.0	-2.7	1.1	-2.9	4171	-0.6	-0.2	**7.0**	**8.9**	**0.6**	**7.8**

(b) Data set: *Mammography*

id	v1	v2	v3	v4	v5	v6	id	v1	v2	v3	v4	v5	v6
SCiForest							iForest						
8901	**31.5**	**3.6**	-0.6	-0.9	-0.4	-0.9	*3336	**6.2**	1.6	-0.6	**9.6**	**23.6**	1.4
*3336	**6.2**	1.6	-0.6	**9.6**	**23.6**	1.4	*5571	**9.2**	1.5	-0.5	**6.3**	**5.0**	0.2
8957	2.5	**4.9**	-0.5	-0.9	-0.4	-0.9	*2222	**3.5**	1.1	-0.6	2.1	**14.4**	0.3
4549	0.4	**5.0**	0.6	-0.9	-0.4	-0.9	8901	**31.5**	**3.6**	-0.6	-0.9	-0.4	-0.9
6035	2.1	**5.0**	-0.1	-0.9	-0.4	-0.9	3608	**14.2**	-0.1	-0.6	**7.9**	-0.4	0.7
9947	1.3	**5.1**	0.8	-0.9	-0.4	-0.9	*10047	**3.9**	0.9	-0.5	2.3	**10.7**	0.7
4838	0.9	**4.9**	0.4	-0.9	-0.4	-0.9	*2220	**6.2**	0.6	-0.5	**9.6**	-0.4	1.3
5538	0.7	**5.0**	0.4	-0.9	-0.4	-0.9	*7823	**2.3**	0.2	-0.5	**3.0**	**13.8**	1.0
8377	0.8	**5.0**	0.5	-0.9	-0.4	-0.9	*3338	1.8	1.0	-0.5	**3.0**	**13.8**	0.7
3823	1.6	**5.0**	-0.1	-0.9	-0.4	-0.9	*1099	**3.0**	1.1	-0.5	2.2	**9.7**	0.7

Table 5.1: SCiForest performs better on data with clustered anomalies and has a higher hit rate in the *Annthyroid* data. However in some data sets, the outlying clustered points are not necessarily anomalies, and iForest is more suitable for these data sets. iForest performs better on data with scattered anomalies and has a higher hit rate in the *Mammography* data. The top ten identified anomalies are presented with their z-scores which measure their deviation from the mean values. Z-scores > 3 are boldfaced and indicate the outlying values. $*$ denotes a ground truth anomaly. Instances with similar boldfaced z-scores imply clustered anomalies.

5.4 Behavioural difference between SCiForest and iForest

In this section, we illustrate SCiForest's behaviour in targeting the clustered anomalies and iForest's behaviour in targeting the scattered anomalies, examples are presented in Table 5.1 in which the feature-value z-scores of the top 10 anomalies are listed for two data sets: *Annthyroid* and *Mammography*. Information of these natural data sets can be found in Appendix A.1. Note that SCiForest targets clustered anomalies, which can be observed by the presence of consistent z-scores in some of the attributes of the top-n identified anomalies. For *Annthyroid* and *Mammography*, high consistent z-scores can be

found in attributes $t3$ and $v2$, respectively. The true anomalies are labelled by "$*$" in Table 5.1. We find that SCiForest has a better detection performance with the *Annthyroid* data set, and iForest is better with the *Mammography* data set. This is because *Annthyroid* has mostly clustered anomalies and *Mammography* has mostly scattered anomalies.

Whether anomalies are scattered or clustered is a domain dependent issue and there are legitimate anomalies in both cases. One of the reasons for having two variants, i.e., iForest and SCiForest, is to be able to choose between the two for different circumstances.

5.5 Sensitivity of the two parameters t and ψ

The performances of iForest and SCiForest are stable in a wide range of t (the number of trees used). Using the two data sets of the highest dimensions, Figure 5.7 shows that AUC converges at a small t for both SCiForest and iForest. The results for iForest on all the natural data sets are available in Appendix C.1 and the results for SCiForest in Appendix C.4. Since increasing t also increases the processing time, the early convergence of AUC suggests that the execution time can be further reduces if t is tuned to a data set.

The performances of iForest and SCiForest are stable in a wide range of sub-sampling sizes for most data sets. In Section 3.2 of Chapter 3, we discovered that a small sub-sampling size improves the detection performance under swamping and masking. In Figure 5.8, using the two largest data sets, *Http* and *ForestCover*, we examine the effect of the sub-sample size on the detection accuracy and processing time on natural data sets. We adjust the sub-sampling size in the range of $\psi = 2, 4, 8, 16, ..., 32768$. We observe that AUC converges very quickly at a small ψ in the *Http* data set and the AUC is near optimal when $\psi > 128$ for iForest and throughout the entire range for SCiForest. With the *ForestCover* data set, the detection performances converge at $\psi = 4096$ under iForest, which is larger than usual. Since *ForestCover* is derived from a geological survey (Blackard, 1998), which contains arbitrary patterns, we conjecture that these arbitrary patterns create difficulty, which requires a higher sub-sampling size to overcome. In Section 5.6, we will examine this condition using four artificially generated data sets.

The results of varying ψ for iForest and SCiForest on all the natural data sets can be found in Appendices C.2 and C.5, respectively. In general, we find that for both SCiForest

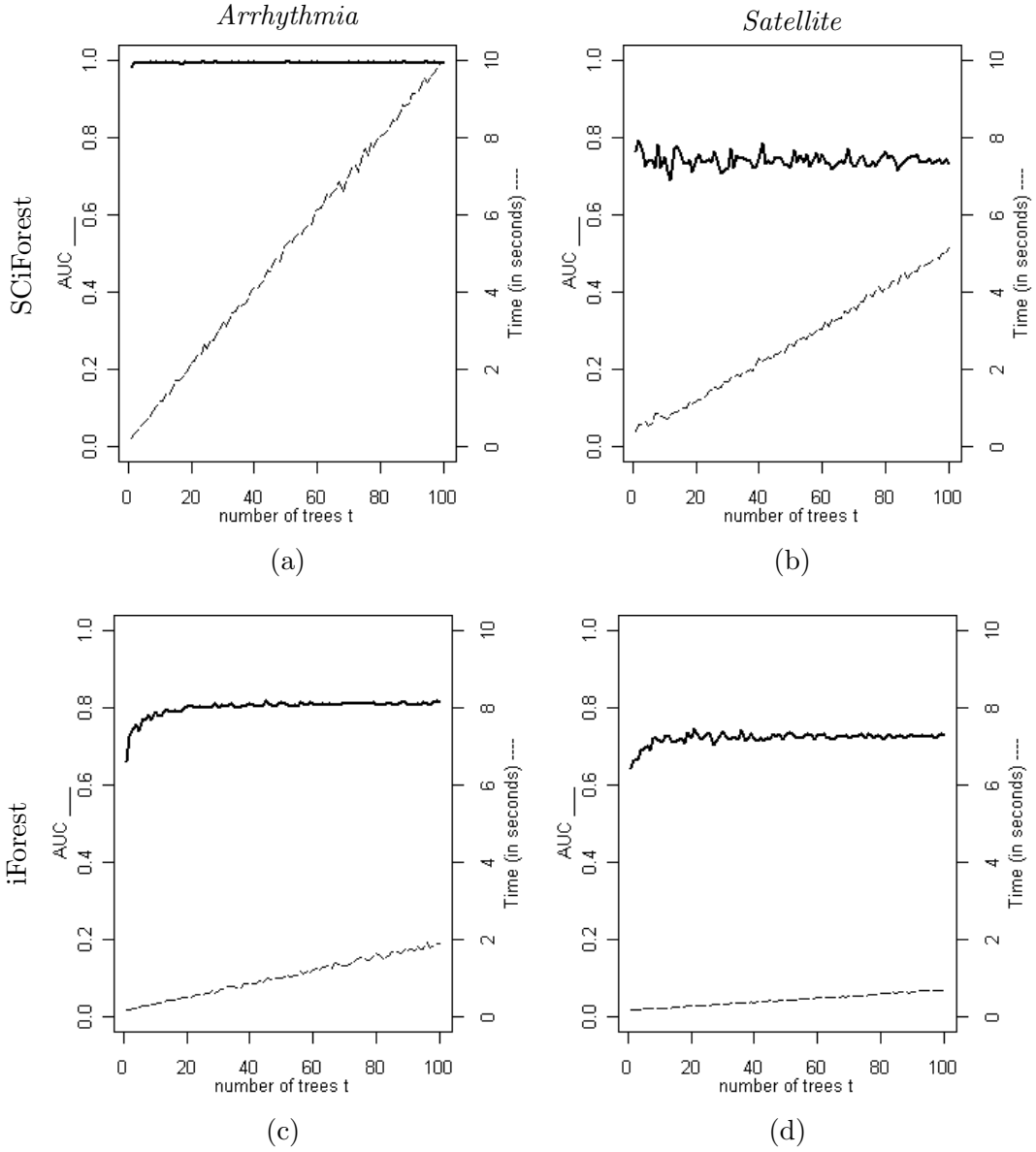


Figure 5.7: In all four diagrams, AUC (y-axis) converges at a small number of trees t (x-axis) with the sub-sampling size ψ set to 256. (a) and (b) are SCiForest's detection performances on *Arrhythmia* and *Satellite*. (c) and (d) are iForest's detection performances on the same data sets.

and iForest, the larger the training sample ψ, the better the detection performance. However, we find that for most of the natural data sets used in this thesis, a small ψ already provides a high AUC, and a further increase of ψ is not necessary.

The processing time increases quadratically when ψ increases from 4 up to 8192 for iForest while maintaining its near optimal detection performance. Along the same lines, SCiForest has a slightly higher increase in the processing time due to the use of hyperplanes and the deterministic split-point selection criterion.

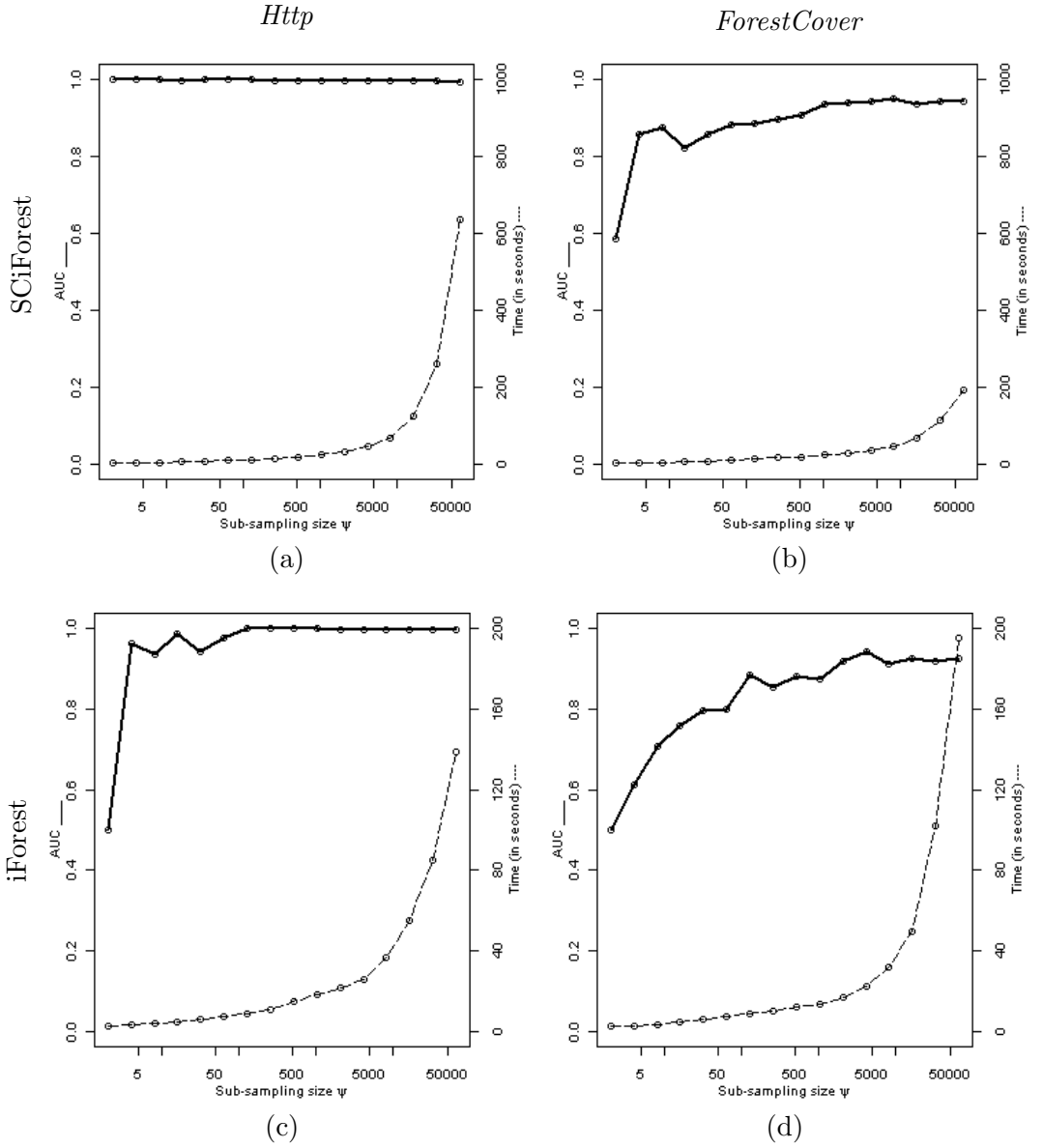


Figure 5.8: Examples of using a small sub-sampling size to achieve a high AUC and low processing time. AUC is shown by the solid lines on the left y-axis and the processing time (in seconds) is shown by the dashed lines on the right y-axis. The sub-sampling size settings (x-axis, log scale) are $\psi = 2, 4, 8, 16, ..., 32768$ with the number of trees constant at $t = 100$. (a) and (b) are SCiForest's detection performances on *Http* and *ForestCover*. (c) and (d) are iForest's detection performances on the same data sets.

In this section, we have demonstrated that the detection performances of the isolation-based methods are stable in a wide range of t and ψ. The detection performance using the default settings is not very different from the detection performance over a wide range of t and ψ for most data sets. However, an incentive to optimize these parameters is to reduce the processing time. Our result also shows that a similar detection performance with a lower processing time can be achieved with smaller values of t and ψ.

5.6 Detecting Anomalies among Arbitrary Patterns

In this section, we utilize four artificial data sets[2] to illustrate the ability of iForest and SCiForest to detect anomalies among arbitrary patterns. We also find that they require a higher sub-sampling size to do so. These four synthetic data sets are shown in Figure 5.9 and their properties are listed in Table 5.2. All the data sets have very dense normal instances and scattered anomalies. Clustered anomalies are not present in these data sets.

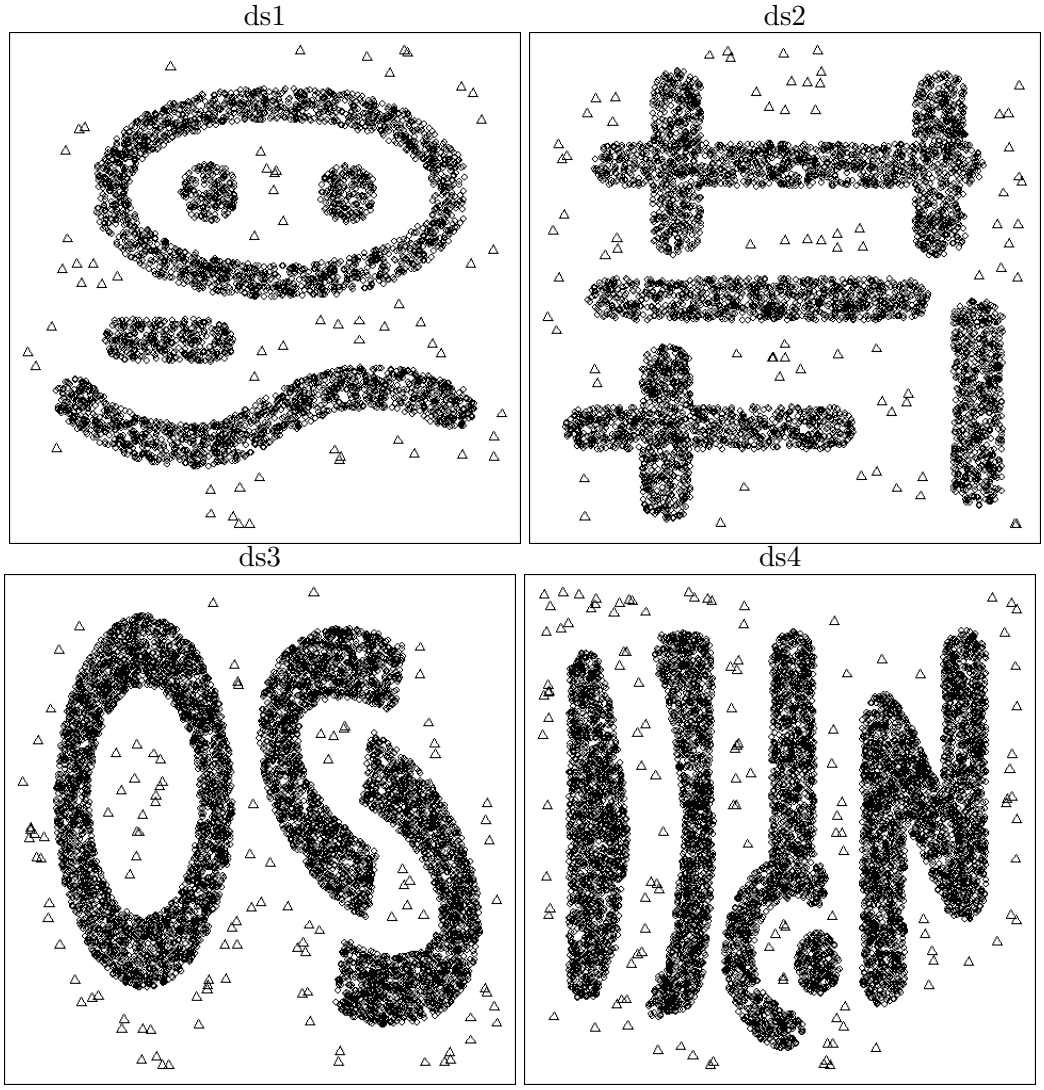


Figure 5.9: Visualization of the arbitrary pattern data sets. The circles denote normal points and triangles denote anomalies

In Figure 5.10, we examine the detection performance (AUC) of the isolation-based methods over a range of sub-sampling sizes $\psi = \{256, 512, 1024, ..., 8192\}$. Under all four data sets, iForest's detection performance increases with an increased sub-sampling size.

[2]These data sets were made available by the courtesy of Xiao Yu. They are similar to those used in (Yu et al., 2009).

Data set	n	m
ds1	4060	60
ds2	5075	75
ds3	7105	105
ds4	8120	120

Table 5.2: Data properties, n denotes the total number of data points and m denotes the number of anomalies.

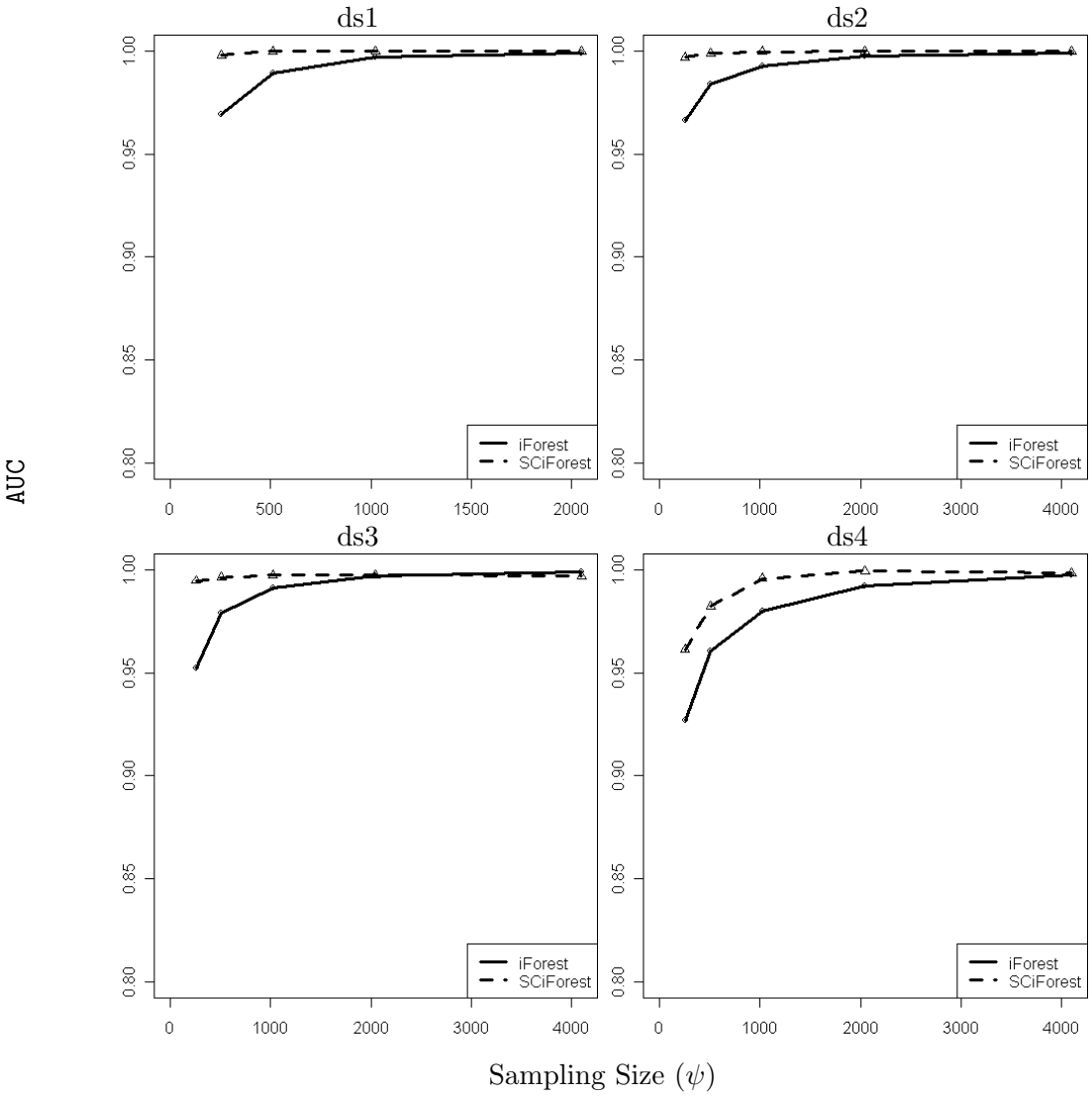


Figure 5.10: The AUC increases as the sampling size increases for both iForest and SCiForest. When the sub-sampling size is 4000, AUC approaches 1.

Generally, SCiForest has a better initial detection performance. We conjecture that this is due to the use of the $Sd_{reduction}$ criterion in SCiForest. For both the isolation-based methods, we find that when a sub-sampling size is small, the anomalies in between the different arbitrary patterns are not being detected. However, when the sub-sampling size increases, the anomalies in between the arbitrary patterns become detectable. This is

because a small sub-sampling size causes the leaf nodes to cover a larger area in the instance space due to the lack of data points for further subdivision. When the areas covered by the leaf nodes are large, they cannot fit inside the space between arbitrary patterns and therefore the anomalies in these leaf nodes are given the path length of their nearby normal instances. When the sub-sampling size increases, the coverage areas of these leaf nodes become small enough to fit inside the space between arbitrary patterns. Under these conditions, the anomalies are given shorter path lengths compared to the normal instances.

When the sub-sampling size increases, the detection performance improves as shown in Figure 5.10. When $\psi = 4000$, the detection accuracy (AUC) of both SCiForest and iForest approach 1. In summary, our results confirm that isolation-based methods are able to detect anomalies among arbitrary patterns and they require a higher sub-sampling size to do so.

5.7 The Utility of Hyper-Planes in SCiForest

When clustered anomalies are present, SCiForest's detection performance increases by selecting better hyper-planes. In the following examples, both the *Annthyroid* and *Dermatology* data sets, are known to have clustered anomalies; however, the *Dermatology* data set has an increasing AUC as q increases while the reverse is true for the *Annthyroid* data set. Both data sets are presented with a performance analysis of SCiForest with various q in comparison with iForest, LOF and ORCA in their default settings.

For *Annthyroid* in Figure 5.11, we find that the anomalies have very different values in attribute "T3" only and thus it is fitting that $q = 1$ performs the best. With the *Dermatology* data set, increasing q has a positive effect on the detection performance of SCiForest. After examining the anomalies, we find that they are different from the normal instances in many attributes and therefore, increasing q improves the detection performance until $q = 4$ when there is no more effect. In summary, when the anomalies are different from normal instances in multiple attributes, the parameter q can be adjusted to optimize the detection performance. The detection performance of SCiForest on other data sets with various q can be found in Appendix C.7. Among all the data sets, *ForestCover*, *Smtp*, *Mammography*, *Pima* and *Dermatology* improve with the increase of q. *Http*, *Shuttle*,

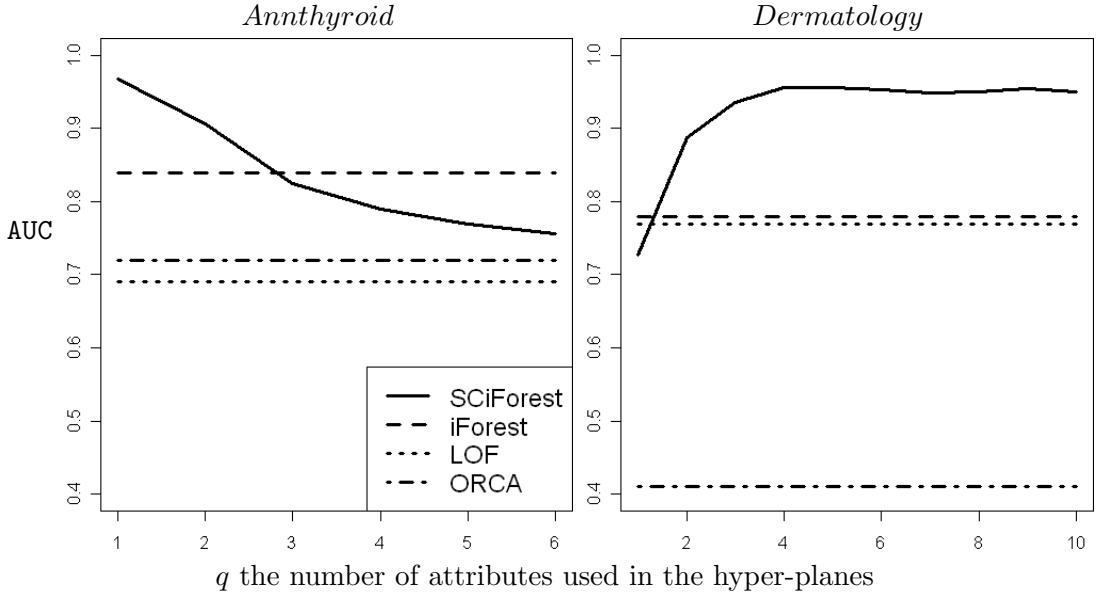


Figure 5.11: Performance analysis on the utility of the Hyper-plane in natural data. The AUC (y-axis) increases with the number of attributes q used in the hyper-planes (x-axis) when anomalies are different in many attributes as in *Dermatology*. In *Dermatology*, the smallest class is defined as anomalies; in Annthyroid classes 1 and 2.

Satellite and *Arrhythmia* are not affected by the different values of q. *Annthyroid*, *Breastw* and *Ionosphere* are affected negatively by an increase in q.

5.8 The Effect of Split Selection Trials in SCiForest

In SCiForest, multiple split selection trials help to seek out better hyper-planes in isolating the clustered anomalies. The number of trials is controlled by the parameter τ. In each of the split selection trials, the maximum $Sd_{reduction}$ is computed for the current hyper-plane. Only the hyper-plane with the best $Sd_{reduction}$ value is retained.

In Figure 5.12, using *Annthyroid* and *Dermatology*—two data sets with known clustered anomalies, we demonstrate that an increase of τ improves the detection performance of SCiForest. However, this improvement plateaus after a threshold, e.g., $\tau > 10$. The results for other data sets can be found in Appendix C.7. Other than *Annthyroid* and *Dermatology*, most of the data sets are not affected by different settings of τ. One exception is *ForestCover*, which has a reduction of detection accuracy with the increase of τ. We conjecture that this data set contains irrelevant attributes, which have higher $Sd_{reduction}$ values as compared to relevant attributes. As the number of trials increases, SCiForest selects more and more irrelevant attributes due to their higher $Sd_{reduction}$ values and hence reduces SCiForest's detection accuracy.

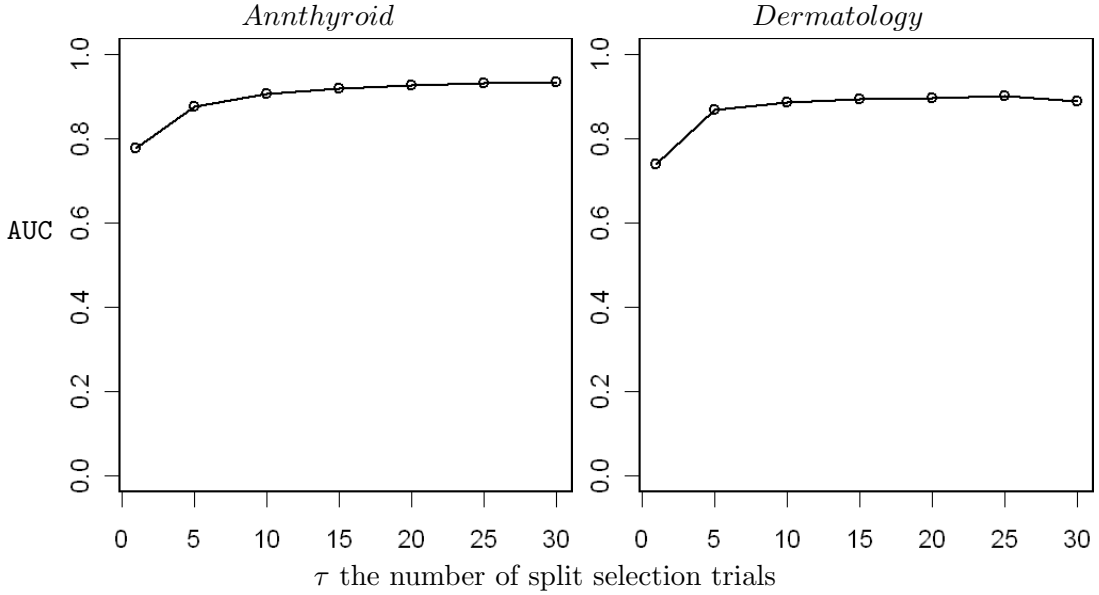


Figure 5.12: For clustered anomalies, the detection performance of SCiForest improves with the increase of τ. AUC (y-axis) increases with τ the number of split selection trials (x-axis).

In summary, τ can be increased to improve the detection accuracy for clustered anomalies. However, this improvement plateaus after a certain threshold.

5.9 Visualizing Anomalies using Isolation Models

It is important to find the attributes that distinguish anomalies. Correct identification of these attributes helps to visualize and interpret the identified anomalies. In this section, we will first look at how iForest utilizes its tree structures to find the best attributes to distinguish the anomalies. Then, we will see how SCiForest visualizes the anomalies with its deterministic split-point selection criterion and hyper-planes.

For illustration purpose, we made use of the *wine* data set from UCI repository (Asuncion and Newman, 2007) and a synthetic data generator (Mulcross) to illustrate how anomalies can be visualized using models from iForest and SCiForest. Information about Mulcross can be found in Appendix A.2. The *wine* data set has 178 instances in 13 attributes. It was first used in (Hodge and Austin, 2004) to illustrate anomalies. Since it has no ground truth anomalies, we can only use it for illustrative purposes.

In iForest, we find that the most frequent last-tested-attributes that isolate an anomaly are the most relevant in terms of the data visualization and interpretation. To explain this correlation, in a collection of iTrees, the last attributes that isolate a point are working

#	Alco.	Malic	Ash	Alcal.	Mag.	Phe.	Flav.	Nonflav.	Pro.	Color	Hue	Dilut.	Proline
122	8	4	**15**	5	6	4	**17**	5	5	11	5	10	5
60	7	8	**15**	**18**	3	5	8	4	9	6	5	8	4
111	12	**13**	8	0	5	8	7	3	**25**	5	5	4	5
159	6	8	11	7	6	**14**	8	4	10	**14**	3	6	3
70	5	6	10	5	**14**	**13**	4	5	9	12	4	9	4
15	6	4	8	5	9	**11**	10	7	**12**	6	6	5	**11**
74	3	4	9	**18**	8	8	5	3	3	**10**	9	10	10
14	**12**	7	4	6	3	11	**12**	**12**	6	8	5	8	6
147	8	7	10	5	**11**	**13**	7	3	9	5	6	8	8
116	19	4	3	8	4	7	0	**12**	8	6	**22**	4	3

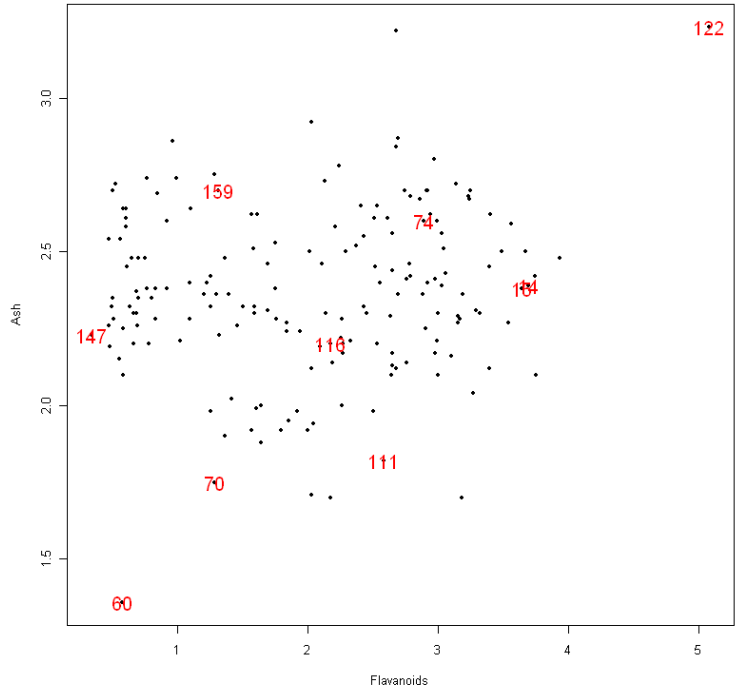
Table 5.3: Matrix A for the top-10 anomalies identified by iForest in the wine data set. Bold faced are the top-two last tested attributes.

in the most local region. Also, when a point is isolated, the tree growing process stops. The frequencies of these attributes reflect the likelihood of these attributes being used to isolate a particular point. Therefore, the last-tested-attributes for a particular anomaly have the power to distinguish it visually.

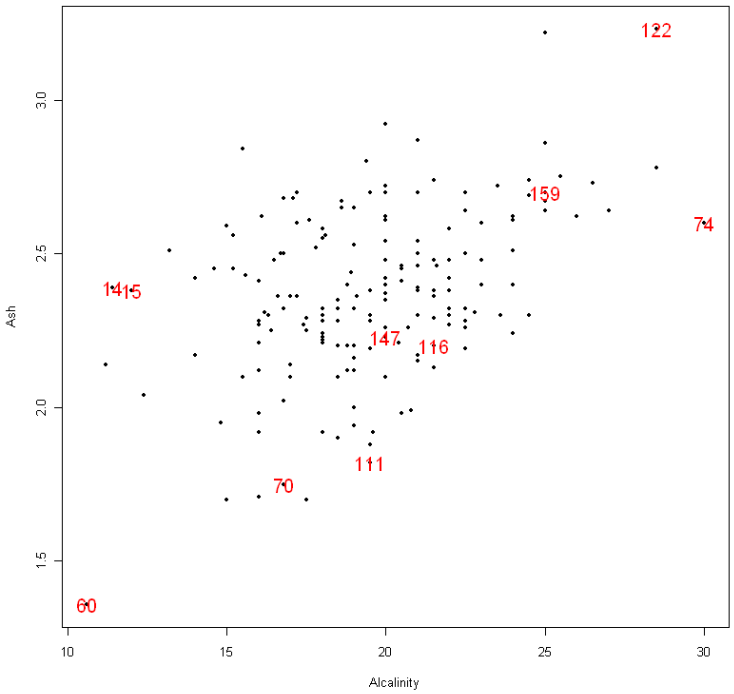
To keep track of these last-tested-attributes, let A be a matrix of size $n \times d$ that is initialized to zero. Matrix A counts the number of times an attribute is the last tested for a particular data point. Simply, for each iTree, when attribute j is the last tested attribute for x, $A[x, j]$ is incremented by 1. After matrix A is updated for each anomaly $\mathbf{x}$ in the matrix A, the attributes with high matrix values are indicates as being the attributes that distinguish $\mathbf{x}$. In Table 5.3, we show an example of this matrix using *Wine* data set.

When inspecting Table 5.3, we find that #122 (the top anomaly) has the highest frequencies in the attributes *Ash* and *Flavanoids*. Using these attributes, it is shown in Figure 5.13 that #122 is indeed far away from the main data cloud located in the upper right hand corner. Thus, the attributes *Ash* and *Flavanoids* are useful for distinguishing and identifying #122. The same also applies to #60, the second anomaly in Table 5.3 and more examples can be found in Appendix D. Note that some of the unmarked outlying points that are shown in Figure 5.13, are only outlying with respect to the attributes that are being used. The marked points are the top ten anomalies identified by iForest, which takes all the attributes into consideration.

Unlike iForest, the deterministic split-point selection criterion and hyper-planes in SCiForest offer a different means to visualize the anomalies. Because of the deterministic nature of the split selection criterion, SCiForest often selects a split that separates most of the anomalies from normal instances at the root node of an iTree. Therefore, the

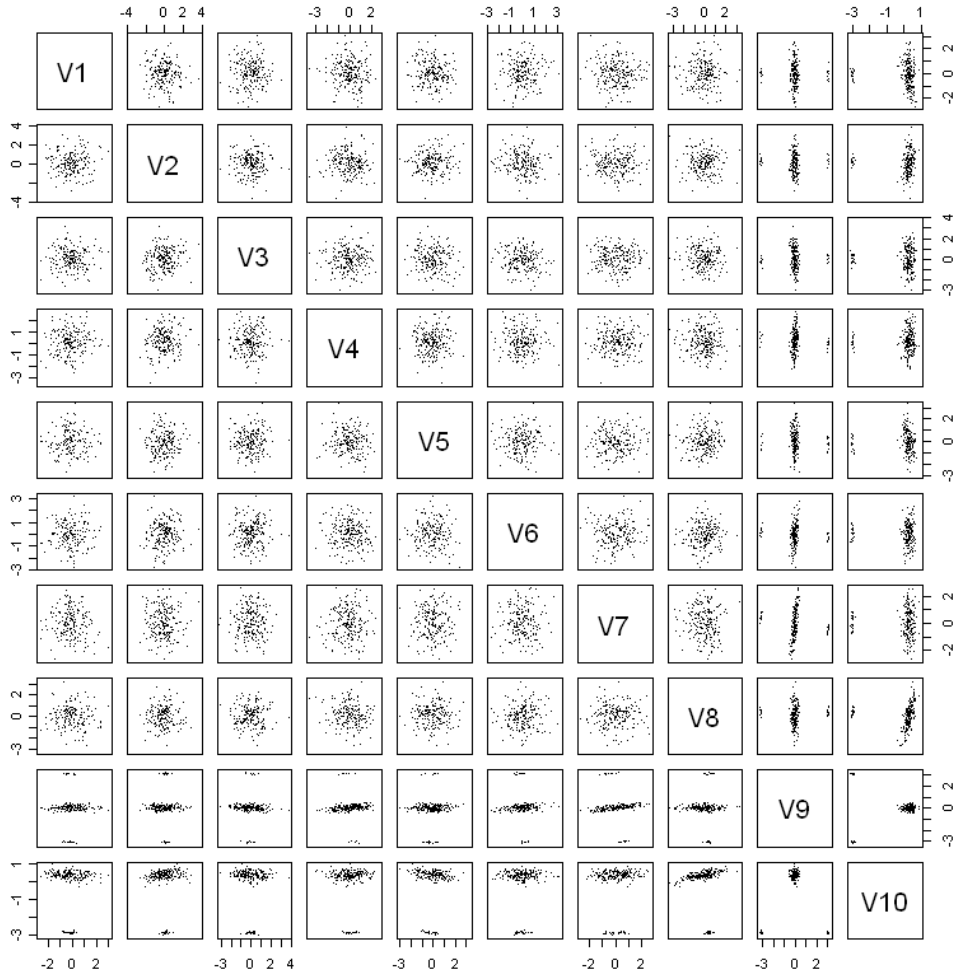


(a) #122 (upper right corner) is distinguishable using attributes *Ash* and *Flavanoids*.



(b)#60 (lower left corner) is distinguishable using attributes *Ash* and *Alcalinity*.

Figure 5.13: To demonstrate iForest's ability to visualize scattered anomalies, the *wine* data set is plotted using the attributes suggested in Table 5.3 to visualize each anomaly. The data point #122 is distinguishable with the attributes *Flavanoids* and *Ash* and data point #60 is distinguishable with the attributes *Alcalinity* and *Ash*.



(a)

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
3.2	1.2	2.3	3.8	3.5	4.8	1.8	2.0	**47.9**	**23.8**

(b)

Figure 5.14: (a) shows a multiple-plot of an artificially generated data set (Mulcross $d = 10, D = 20, cl = 2, n = 500, a = 0.1$). Clustered anomalies can be distinguished using attributes V9 and V10. (b) shows the values of array B updated from all the root nodes in a SCiForest model. Note that V9 and V10 have the highest values (boldfaced) in (b), which corresponds to their ability to visualize the clustered anomalies shown in (a).

root nodes in SCiForest contain the information needed to visualize the anomalies. Since SCiForest uses hyper-planes instead of individual attributes to separate the data, we have to extract the information from the attribute's coefficients of the hyper-planes to visualize the anomalies. Our innovation is to find the most involved attributes in root nodes by summing the absolute values of an attribute's coefficients among all the root node hyper-planes. Simply, let B be an array of size d, i.e. $|B| = d$, $B = \{b_1, ..., b_d\}$, initialized to

zero. For each root node in a SCiForest's model, b_i is incremented with the term $abs(C_i)$, which is the absolute value of the hyper-plane coefficient of attribute i. After array B is updated, the elements with the highest values indicate the attributes that distinguish the anomalies.

Using the Mulcross data generator, we create a situation in which only two of the ten attributes are relevant to the anomalies. The Mulcross data generator is set as follows: $d = 10, D = 20, cl = 2, n = 500, a = 0.1$. This setting creates two anomaly clusters of 50 points in addition to one cluster of normal instances. They are illustrated in Figure 5.14(a). The B array extracted from SCiForest's model is reported in Figure 5.14(b). We can see that the attributes V9 and V10 have exceptionally high values and are able to visualize the clustered anomalies as shown in Figure 5.14(a).

In Section 5.4, we found that iForest targets scattered anomalies and SCiForest targets clustered anomalies. These characteristics also affect their ability to visualise the anomalies. When implementing iForest's model on Mulcross's data, the most frequent last-tested-attributes are not able to distinguish the clustered anomalies. This is because the last-tested-attributes are used to separate an anomaly from the other anomalies rather than the normal points. In iForest, the attributes of the root nodes are randomly assigned and their frequencies are uncorrelated to the attributes that distinguish the anomalies.

Likewise, when implementing SCiForest's model on the $Wine$ data set, array B cannot be used to highlight the attributes that distinguish the individual anomalies. The highest elements in B correspond to the attributes $Nonflavanoids$ and Hue. Also, the attribute involved in the last-test-nodes is uncorrelated to the attributes that distinguish the anomalies. For example, instance #60 of the wine data set has the highest attribute involvements in $Nonflavanoids$ and $Flavanoids$, which are not helpful in visualizing #60.

In summary, iForest visualizes the scattered anomalies using the most frequent last-tested-attributes. On the other hand, SCiForest visualizes the clustered anomalies using the most involved attributes at the root nodes.

5.10 Chapter Summary

We have examined many characteristics of iForest and SCiForest. Their common characteristics are:

- They are able to detect anomalies in many situations, including: (i) when the anomalies have varying densities, (ii) when both scattered and clustered anomalies are present and (iii) when the normal points have varying densities.
- They are able to detect anomalies among arbitrary patterns, using a higher subsampling size.
- They have a stable performance under a wide range of parameters t and ψ for most data sets.

Their different characteristics are:

- SCiForest is able to detect a very difficult class of anomaly known as local clustered anomalies.
- SCIForest has better breakdown properties in terms of (a) size-increasing clustered anomalies and (b) anomalies that are very close to normal points.
- Parameters q and τ in SCiForest are insensitive in a wide range of different settings. However, they can be adjusted to improve the detection performance for (i) anomalies that depend on multiple attributes and (ii) clustered anomalies.
- iForest can be used to visualize scattered anomalies and SCiForest for clustered anomalies.

Chapter 6

Empirical Evaluation

This chapter presents detailed results for experiments designed to evaluate iForest and SCiForest, and compares them to three state-of-the-art anomaly detectors, i.e., ORCA, LOF and SVM in terms of the detection performance and processing time.

The organization of this chapter is as follows: Section 6.1 describes the experimental settings used in this chapter. Section 6.2 features a comparison of five anomaly detectors including SCiForest, iForest, ORCA, LOF and SVM, in terms of the detection performance and processing time using twelve natural data sets. Section 6.3 evaluates iForest's and SCiForest's efficiencies with the Mulcross data generator. We generate up to one million data points to find how the isolation-based methods scale compared to other anomaly detectors. Section 6.4 shows the results of iForest and SCiForest when they are trained without anomalies. Finally, Section 6.5 summarizes this chapter.

6.1 Experimental Setting

As for the performance analysis and evaluation, twelve natural data sets and a synthetic data generator are used. These are selected because they contain known anomaly classes and have been used in the literature to evaluate anomaly detectors in similar settings. These twelve natural data sets include: the two biggest data subsets (*Http* and *Smtp*) of the KDD CUP 99 network intrusion data used in (Yamanishi et al., 2000), *Annthyroid*, *Arrhythmia*, *Dermatology*, Forest Cover Type (*ForestCover*), *Ionosphere*, *Pima*, *Satellite*, *Shuttle*, Wisconsin Breast Cancer (*Breastw*) (Asuncion and Newman, 2007) and *Mammography*[1]. Details of these data sets can be found in Appendix A.1. It is assumed that

[1]The Mammography data set was made available by the courtesy of Aleksandar Lazarevic

the anomaly labels are unavailable at the training stage. The anomaly labels are only available at the evaluation stage to calculate the detection performance measures. Since, the experiments are only concerned with continuous-valued attributes, all nominal and binary attributes are removed.

In terms of the anomaly detectors, the settings of SCiForest and iForest can be found in Table 4.3 in Chapter 4. ORCA, LOF and SVM are all used for comparison. For ORCA, the parameter k determines the number of nearest neighbours and the parameter N determines how many anomalies are reported. Using ORCA's default setting ($k = 5$ and $N = 30$), all data sets larger than one thousand points report an AUC close to 0.5, which is equivalent to not using any anomaly detector. We use ORCA's default setting of $k = 5$ in our experiments. However, we also use a reasonable value of $N = \frac{n}{8}$ for the experiments in this chapter to get a decent detection performance. In reporting ORCA's processing time, we report the total training and testing time, but omit the pre-processing time "dprep", which randomizes the sequence of instances. For LOF, we use the commonly used setting of $k = 10$ for our experiments. For one-class SVM, we use the Radial Basis Function kernel and inverse width parameter as suggested in (Caputo et al., 2002). The implementation details of these anomaly detectors can be found in Appendix A.4.

In this chapter, the actual CPU time and Area Under Curve (AUC) are reported. We also report the H measure[2] (Hand, 2009) for major results in addition to the AUC. The H measure is an alternative to AUC; it is argued that in some situations, AUC differs from H measure, e.g., when the ROC curves are crossed (Hand, 2009). We expect the AUC and H measure to provide similar results in our experiments, since we observe that ROC curves in anomaly detection are rather simple, which avoids the above-mentioned situation. The rationale for using the AUC and H measure is that we cannot know the definite number of anomalies in the data. Hence, the AUC and H measure are suitable for this evaluation, since they are summary measures for the detection rate without using a fixed threshold. For algorithms that consist of random elements that affect the detection performance, their results are reported using the average of ten runs. These algorithms are iForest and SCiForest.

The experiments in this chapter are conducted as single-threaded jobs processed at 2.3GHz in a Linux cluster (www.vpac.org).

[2] `http://www2.imperial.ac.uk/~djhand/`

6.2 Comparing Five Anomaly Detectors

	AUC				
	iForest	SCiForest	ORCA	SVM	LOF
Http	**1.00**	**1.00**	0.36	0.90	NA
ForestCover	0.87	0.71	0.83	**0.90**	0.57
Smtp	**0.90**	0.87	0.80	0.78	0.32
Shuttle	**1.00**	**1.00**	0.60	0.79	0.55
Mammography	**0.87**	0.61	0.77	0.65	0.67
Annthyroid	0.84	**0.91**	0.69	0.63	0.72
Satellite	0.72	**0.74**	0.65	0.61	0.52
Pima	0.67	0.65	**0.71**	0.55	0.49
Breastw	**0.98**	**0.98**	**0.98**	0.66	0.37
Arrhythmia	**0.81**	0.73	0.78	0.71	0.73
Dermatology	0.78	**0.89**	0.77	0.74	0.41
Ionosphere	0.84	0.91	**0.92**	0.71	0.90
	wins/draws/loses				
iForest		5/3/4	9/1/2	11/0/1	11/0/0
SCiForest			6/1/5	10/0/2	9/1/1

Table 6.1: The performance of five anomaly detectors in terms of the AUC. Best scores are boldfaced. 'NA' denotes a process run of more than two weeks.

	H measure				
	iForest	SCiForest	ORCA	SVM	LOF
Http	**0.895**	0.423	0.009	0.001	NA
ForestCover	0.003	**0.011**	0.003	0.004	0.000
Smtp	0.000	0.000	0.000	0.000	0.000
Shuttle	0.818	**0.896**	0.046	0.077	0.004
Mammography	**0.065**	0.054	0.047	0.019	0.000
Annthyroid	**0.115**	0.048	0.038	0.013	0.032
Satellite	0.292	**0.304**	0.102	0.078	0.010
Pima	0.096	0.051	**0.162**	0.016	0.008
Breastw	0.807	**0.872**	0.851	0.236	0.004
Arrhythmia	0.231	**0.236**	0.184	0.129	0.107
Dermatology	0.076	**0.106**	0.025	0.025	0.001
Ionosphere	0.478	0.504	**0.737**	0.163	0.621
	wins/draws/loses				
iForest		4/1/7	7/2/3	10/1/1	9/1/1
SCiForest			9/1/2	11/1/0	9/1/1

Table 6.2: The H performance measures of the five detectors. The best scores are boldfaced. SCiForest has six out of the twelve best scores, followed by iForest three and ORCA two. The H measure reports a similar result to the AUC in Table 6.1. Note that *Smtp* has very small H measures across all five detectors.

The aim of this experiment is to compare iForest and SCiForest with ORCA, LOF and SVM in terms of the AUC and processing time using twelve natural data sets. Table 6.1 reports the AUC score and Table 6.3 reports the actual run time of all methods. Table 6.2

	Time (seconds)				
	iForest	SCiForest	ORCA	SVM	LOF
Http	14.13	39.22	9487.47	34979.76	NA
ForestCover	11.18	38.25	6954.93	9443.91	224380.19
Smtp	2.84	10.35	266.87	986.98	24280.65
Shuttle	2.53	6.69	158.92	331.11	7961.38
mammography	0.46	1.15	1.54	10.69	15459.82
Annthyroid	0.39	5.91	2.39	4.17	121.58
Satellite	0.74	5.38	8.97	9.13	528.58
Pima	0.21	1.10	2.08	0.06	1.50
Breastw	0.21	1.16	0.04	0.08	2.14
Arrhythmia	1.79	11.81	0.48	0.15	23.72
Dermatology	0.27	1.04	0.04	0.04	0.91
Ionosphere	0.28	4.43	0.04	0.04	0.96

Table 6.3: The performance of five anomaly detectors in terms of the processing time (training time + evaluation time). 'NA' denotes a process run of more than two weeks.

reports the same result using the H measure. From Table 6.1, we observe that iForest and SCiForest compare favourably to all other methods. The wins/draws/loses count can be found at the bottom of Tables 6.1 and 6.2. When comparing iForest and SCiForest, the result suggests that they have similar detection performance on these natural data sets. Using the AUC, iForest has five wins, three draws and four loses against SCiForest. Using the H measure, iForest has four wins, one draw and seven loses against SCiForest.

When comparing iForest and SCiForest to all the other methods, we find that iForest and SCiForest have the majority wins in both the AUC and H measure. Considering that there are twelve data sets in total, iForest and SCiForest have nine or more wins over all the other methods, except for SCiForest vs. ORCA in AUC and iForest vs. ORCA with the H measure. This result suggests that iForest and SCiForest compare favourably to the other methods in this evaluation. In the sign test, SCiForest and iForest perform significantly better than LOF and SVM at $\alpha = 0.05$ two-tailed.

In the *Http* data sets, due to the large anomaly-cluster size and the fact that the anomaly clusters have a higher density compared to the normal instances (i.e., the masking effect), ORCA reports a poorer than random performance, i.e. $AUC < 0.5$. We also test ORCA on these data sets using a higher k value ($k = 150$) with a similar detection performance. This highlights a problematic assumption in ORCA and other similar k-nn based methods in that they can only detect low-density anomaly clusters, which have data

size smaller than k. Increasing k may solve this problem, but is not practical in a high volume setting due to the increase in the processing time.

Note that the particular LOF implementation that we used has an extended runtime and could not complete the *Http* data set within two weeks. We have attempted another LOF implementation in ELKI (Achtert et al., 2010) to process the *Http* data set which finished in 70,591 seconds. However, the output LOF values consist of many 'Nan' and 'Infinity' values, which prevent us from calculating any performance measure.

Looking at Table 6.3, both iForest and SCiForest outperform ORCA, SVM and LOF in terms of the processing time for large data sets. In particular, iForest is more accurate and faster in all the data sets larger than one thousand points. SCiForest is a bit slower than iForest due to the additional calculation of $Sd_{reduction}$ and the hyper-planes. However, SCiForest still outperforms ORCA, SVM and LOF in all data sets larger than ten thousand points.

Note that the difference in the execution time is huge when comparing iForest and SCiForest with the other three detectors (ORCA, SVM and LOF) especially in large data sets. For examples, SCiForest requires about one nine-hundredth of SVM's processing time for the *Http* data set; iForest requires about one six-hundredth of ORCA's processing time for the $ForestCover$ data set. This difference is due to the fact that iForest and SCiForest do not compute any distances. This is also despite the fact that ORCA reports only $\frac{n}{8}$ points whereas the isolation-based methods rank all n points. In the following section, we further examine the efficiency of isolation-based methods using a synthetic data generator.

6.3 Efficiency Analysis for Isolation-Based Methods

When using SCiForest and iForest with parameters $\psi = 256, t = 100$ on Mulcross generated data of different sizes, we find that the training times of the isolation-based methods grow slightly less than 50% when the size of the data doubles as reported in Table 6.4. Discounting the operation of loading the training data into the memory and the sampling process, the time taken to construct $t = 100$ trees is constant at approximately 0.04 seconds (iForest) and 0.08 seconds (SCiForest). The total processing times for the isolation-based methods grow of the same rate as the data size. In contrast, both ORCA and SVM have their execution times quadruple when the data size doubles. There is a

significant difference when the data size is very large, e.g., with one million data points, the isolation-based methods take less than one eight-hundredth of the time that ORCA takes and one three-thousandth of that of SVM; and the detection accuracies are a lot better too. The above results show that the isolation-based methods are significantly more efficient compared to ORCA and SVM in large data sets using small sub-samples. It is unclear how other methods can make use of small sub-samples to speed up their runtime.

	AUC				Time (seconds)			
n	iForest	SCiForest	ORCA	SVM	iForest	SCiForest	ORCA	SVM
1024	0.94	0.94	0.23	0.59	0.11	0.14	0.08	0.10
2048	0.92	0.93	0.28	0.60	0.12	0.15	0.21	0.68
4096	0.87	0.87	0.31	0.59	0.17	0.29	0.73	2.61
8196	0.94	0.94	0.33	0.49	0.26	0.47	2.63	9.59
16384	0.93	0.93	0.33	0.58	0.43	0.85	10.19	37.04
32768	0.89	0.90	0.33	0.59	0.77	1.58	40.20	152.82
65536	0.93	0.93	0.33	0.59	1.62	3.09	158.16	551.88
131072	0.90	0.90	0.33	0.59	3.20	6.12	637.83	2040.63
262144	0.91	0.91	0.33	0.59	6.20	12.22	2535.60	7899.46
524288	0.90	0.91	0.33	0.49	12.32	24.40	10203.64	28737.42
1048576	0.88	0.89	0.33	0.59	24.26	48.12	40779.85	155084.49

Table 6.4: Performance of iForest, SCiForest, ORCA and SVM on Mulcross ($n = \{1024, ..., 1048576\}, d = 4, cl = 2, D = 2, a = 0.1$) with various data sizes.

6.4 Training Using Normal Instances Only

"Do iForest and SCiForest work when the training set contains normal instances only?" To answer this question, we remove the anomalies from the training sample and evaluate the trained model with both the anomalies and normal instances. The AUC results of training with and without the anomalies are reported in Table 6.5. The results show that the exclusion of the anomalies does not degrade the detection performances of both SCiForest and iForest, with one exception in the *ForestCover* data set.

With the anomalies absence from the training data, the reason why the isolation-based methods are still be able to detect anomalies is that the isolation-based methods assign a short path length to any point located in regions with no training instances. Thus, the absence of the anomalies in the training sample does not affect the detection performance—one less thing to worry about when using the isolation-based methods.

Training SCiForest with only the normal instances produces a similar result as training with both the normal instances and anomalies. The *ForestCover* data set is the only data

	Normal instances only		Normal instances and anomalies	
	SCiForest	iForest	SCiForest	iForest
Http	0.99	1.00	1.00	1.00
ForestCover	0.59	0.87	**0.71**	0.87
Smtp	0.87	0.90	0.87	0.90
Shuttle	1.00	1.00	1.00	1.00
Mammography	**0.65**	**0.88**	0.61	0.87
Annthyroid	**0.97**	**0.92**	0.91	0.84
Satellite	**0.83**	**0.82**	0.74	0.72
Pima	**0.67**	**0.78**	0.65	0.67
Breastw	**0.99**	**1.00**	0.98	0.98
Arrhythmia	**0.75**	**0.85**	0.73	0.81
Dermatology	**1.00**	**0.97**	0.89	0.78
Ionosphere	0.91	**0.96**	0.91	0.84

Table 6.5: The AUC performance of iForest trained with normal instances only as compared with training with both the normal instances and anomalies. Better performances are boldfaced.

set that is affected by the lack of anomalies. We suspect that the *ForestCover* data set contains a complex data structure which misleads $Sd_{reduction}$ and the presence of the anomalies is needed in order for $Sd_{reduction}$ to select proper hyper-planes that separate the anomalies from the normal points.

In summary, both SCiForest and iForest work well with or without anomalies present in the training samples.

6.5 Chapter summary

The focus of this chapter is a performance evaluation of the two isolation-based methods iForest and SCiForest compared to three state-of-the-art anomaly detectors, ORCA, LOF and SVM. Using the twelve natural data sets, both the AUC and H measure show that our isolation-based methods have significantly better detection performance as compared to the other three detectors. We also demonstrate the efficiency of the isolation-based methods, which is significantly better than ORCA and SVM. We show that our isolation-based methods are able to detect the anomalies regardless of whether or not the anomalies are present in the training data sets.

Chapter 7

Behaviours on High Dimensional Data

Handling high dimensional data is an important issue in anomaly detection. In distance-based and density-based methods, every point becomes equally sparse in the high dimensional space, making the distance a meaningless measure. Note that many density-based methods also use the distance to estimate the density. This phenomenon is referred to the 'curse of dimensionality' and causes difficulties for most methods to handle the high dimensional data without some kind of dimensionality reduction scheme.

The aim of this chapter is to examine the behaviours of isolation-based methods and explore how they can cope with the high-dimensional data. Experiments in this chapter are not exhaustive, but they do provide ideas on how to make use of isolation-based methods in handling high dimensional data.

Isolation-based methods also suffer from the 'curse of dimensionality'. To combat this, we explore the use of an attribute selector in Section 7.1, to enhance the detection performance of the isolation-based methods—especially for SCiForest, whose deterministic criterion naturally acts as an attribute selector. In Section 7.2, we evaluate SCiForest's detection performance on high dimensional data. In Section 7.3, we examine the conditions under which SCiForest would perform well on high dimensional data. In Section 7.4, we evaluate the effect of an increasing dimensionality using a synthetic data generator.

7.1 Using Attribute Selector

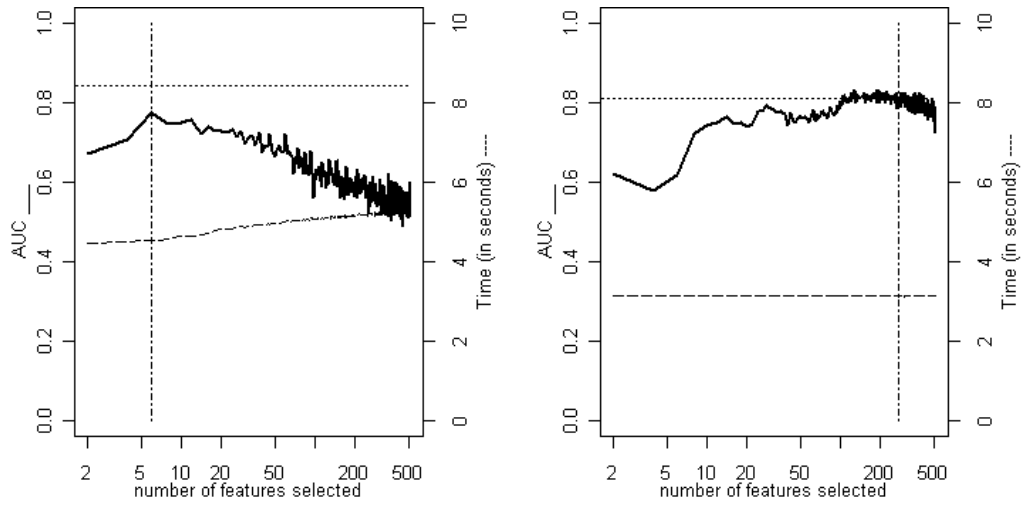
In high dimensional space, the effectiveness of isolating anomalies decreases as data points become sparse or selected attributes are not relevant in isolating anomalies. The aim of this section is to demonstrate iForest's ability to improve its detection performance in high dimensional data by using a simple attribute selector.

In this experiment, we study cases in which the anomalies have a large number of irrelevant attributes in high dimensional space. We simulate high dimensional data using natural and synthetic data sets. Information about these natural data sets can be found in Appendix A.1. We generate a synthetic data set using the Mulcross generator($n = 262144, d = 4, cl = 2, a = 0.1, D = 2$). Information about the Mulcross generator can be found in Appendix A.2. For each data set, a number of uniformly distributed random attributes, valued between 0 and 1 are added, so that there are 512 attributes in total in each data set. Without using any dimensionality reduction scheme, the detection accuracies (AUC) of iForest are close to 0.5 in most data sets (see Table 7.2). Thus, we aim to find out if an additional attribute selector can improve the detection performance of iForest.

With isolation-based methods, there are at least two approaches that can be used to apply an attribute selector: 1) the attribute selector can be applied to each individual sub-sample that is selected before building an iTree, 2) the attribute selector can be applied to the entire data set and only those selected attributes are involved in the training process. The advantage of the first approach is that it saves time. When the size of the training data set is large, applying an attribute selector to small sub-samples can be more cost effective than applying it to the entire data set. The advantage of the second approach is that the attribute selector has complete coverage of all anomalies in the training data set, so as to make a better selection of the relevant attributes. In this section, we will experiment with both approaches in turn.

In our first experiment, we use a simple statistical test, Kurtosis (Joanes and Gill, 1998), to select an attribute subspace from the sub-sample before constructing each single tree in the isolation-based models. In our first experiment, the Kurtosis is applied on each sub-sample before building an iTree in iForest. This reduces the processing time compared to processing the entire data set. To compute the Kurtosis, at least two passes of the entire

data set are needed. The Kurtosis is defined as $\frac{\sum(x-\bar{x})^4}{n\sigma^4} - 3$, where $\bar{x}$ is the mean value of x and σ is the standard deviation. It measures the 'peakedness' of a univariate distribution. It is sensitive to the presence of anomalies and hence is a good attribute selector for anomaly detection. After the Kurtosis has provided a ranking for each attribute, a subspace of the attributes is selected according to this ranking to construct an iTree. The results of this process are promising and we show that the detection performance improves when the subspace size approaches the original number of attributes. There are other attribute selectors available, e.g., Grubb's test (Snedecor and Cochran, 1989, pp. 278-280). However, in this section, we only concern with showcasing iForest's ability to work with an attribute selector to reduce the dimensionality of the anomaly detection tasks. SCiForest has an internal mechanism to deal with high dimensional data, which has a similar effect as applying the attribute selector on small sub-samples, which will be discussed in Section 7.2. The results from other detectors are omitted since most of their detection results are close to 0.5 in AUC without any attribute selector and there is no easy way to apply the Kurtosis to multiple small sub-samples in distance- or density-based methods. For SCiForest, ORCA and iForest, the results of applying an attribute selector on the entire data set will be presented later on in this section.



(a) *Mammography (6 original attributes)* (b) *Arrhythmia (274 original attributes)*

Figure 7.1: iForest achieves good results in high dimensional data when using the Kurtosis to select attributes. Irrelevant attributes are added so that the total number of attributes reaches 512. AUC (left y-axis, solid lines) improves when the subspace size (x-axis) comes close to the number of original attributes, and the processing time (right y-axis, dashed lines, in seconds) increases slightly as the subspace size increases. iForest trained using the original data has a slightly better AUC (shown as the top dotted lines). The dotted dash lines indicate the original number of attributes.

As an illustration, Figure 7.1 shows that i) AUC peaks when the subspace size is the same as the number of original attributes and ii) the processing time of iForest in Mammography and Arrhythmia remains less than 10 seconds for the whole range of subspace sizes. When ORCA is used on these two high dimensional data sets, not only does it report an AUC close to 0.5; it also takes over one hundred seconds to process these high dimensional data sets. This shows that these high dimensional data sets are challenging, however, iForest is able to improve the detection performance by the simple addition of a Kurtosis test on small sub-samples rather than the entire data set. Given a large data set, e.g. *http*, iForest is only required to compute the Kurtosis of about one-twentieth of the data. The results for other data sets can be found in Appendix C.3. Using the number of original attributes, iForest reports a better detection performance compared to using all 512 attributes, except for the Mulcross data set. For the *ForestCover* and *Ionosphere* data sets, a slightly better detection performance is reported with a lower number of selected attributes. We conjecture that some of the original attributes in these data sets are irrelevant to the task of anomaly detection and using less attributes helps to eliminate these irrelevant attributes.

In our second experiment using an attribute selector, we select the attributes based on the entire training set rather than using small sub-samples. For large data sets, we find that the processing time of Kurtosis alone is greater than the processing time using iForest with kurtosis applied on small sub-samples. For processing the Kurtosis alone, *Http* and *ForestCover* take 73.31 and 44.30 seconds, respectively. Assuming the number of selected attributes is the same as the number of original attributes, the processing times for iForest with the kurtosis applied to small sub-samples are 66.96 and 33.56 seconds for the same data sets. We expect this advantage to be even more obvious in larger data sets. However, this effect diminishes when the training data set is small, for examples, the Kurtosis processing times of the *Pima*, *Breastw*, *Arrthymia* and *Ionsophere* data sets take less than one second.

In terms of the detection performance, when applying the Kurtosis to multiple small sub-samples, the Kurtosis's values are biased compared to those based on the entire training data set (Joanes and Gill, 1998). Table 7.1 reports iForest's detection performance on selecting attributes using (a) small sub-samples versus (b) the entire data set. We also

Kurtosis's treatment	iForest		SCiForest	ORCA
	(a) on small sub-samples	(b) on entire data set	(c) on entire data set	(d) on entire data set
Http	1.00	1.00	1.00	0.36
ForestCover	**0.94**	0.89	0.72	0.82
Mulcross	0.53	**0.95**	0.99	0.33
Smtp	0.72	**0.90**	0.86	0.80
Shuttle	0.99	0.99	1.00	0.60
Mammography	0.75	**0.84**	0.63	0.76
Annthyroid	0.59	**0.84**	0.91	0.69
Satellite	0.64	**0.73**	0.74	0.65
Pima	0.68	0.68	0.65	0.71
Breastw	0.97	0.97	0.98	0.98
Arrhythmia	0.80	0.80	0.73	0.78
Ionosphere	0.84	0.84	0.91	0.92

Table 7.1: The detection performance (AUC) of iForest, SCiForest and ORCA on high-dimensional data. Using the Kurtosis attribute selector, (a) selects the attributes using small sub-samples (iForest); (b), (c) and (d) select the attributes once using the entire data set (iForest, SCiForest and ORCA). It is assumed that the number of original attributes is known. The Kurtosis selects k number of attributes, where k is the number of original attributes. Boldfaced values represent better detection performance using iForest.

included the results from SCiForest (Table 7.1(c)) and ORCA (Table 7.1(d)) for references. For iForest, we find that there is a trade off in the detection performance when selecting attributes using small sub-samples, for example the *Mulcross*, *Mammography*, *Annthyroid* and *Satellite* data sets. Comparing iForest, SCiForest and ORCA using kurtosis applied on the entire data set, the isolation-based methods are better than ORCA. iForest has 9 wins and 3 loses compared to ORCA, and SCiForest has 6 wins, 1 draw and 5 loses compared to ORCA.

Overall, although using the Kurtosis as an attribute selector helps in dealing with high-dimensional data, it is important to know how many attributes are needed in the attribute selection process. Since the anomalies are usually only different from the normal instances in a few attributes and the detection performance in our experiment usually peaks in a range between 5 to 20 attributes (see Appendix C.3), we recommend that the number of selected attributes be set to a value between 5 to 20.

Another concern in this experiment is that the artificially generated irrelevant attributes may not be representative of real life situations. In real-life situations, if the irrelevant attributes score higher than the relevant attributes according to the attribute selector, then these irrelevant attributes must be manually removed to achieve a good detection performance.

7.2 SCiForest's Ability to Handle High Dimensional Data

For SCiForest, the ability to handle high dimensional data is inherent in the split selection process because of the use of the $Sd_{reduction}$ split selection trials in selecting the hyper-planes that separate the anomalies from the normal points. In each of the $Sd_{reduction}$ trials, a different hyper-plane is generated. If anomalies are separable in a particular hyper-plane, then $Sd_{reduction}$ returns a higher value compared to the other hyper-planes in which the anomalies are not clearly separable. After a number of trials, the hyper-plane with the highest $Sd_{reduction}$ is selected to separate the data. This mechanism increases the likelihood for relevant attributes to be involved as they are able to separate the anomalies from the normal points. Table 7.2 reports the detection performance of SCiForest and iForest over twelve data sets with added irrelevant attributes. In general SCiForest has better a detection performance over iForest as reflected in the mean AUC at the bottom of the table. Some of the differences are remarkable, e.g., http, Shuttle and Breastw in which the AUC differences are 0.34, 0.32 and 0.27. In the cases of smtp, pima and arrhythmia, the difference between the two methods is minimal with only 0.03 differences in the AUC.

	SCiForest	iForest
Http	**0.91**	0.56
ForestCover	**0.59**	0.54
Mulcross	**0.77**	0.56
Smtp	0.51	**0.53**
Shuttle	**0.95**	0.63
Mammography	**0.69**	0.54
Annthyroid	**0.59**	0.53
Satellite	**0.67**	0.60
Pima	**0.54**	0.52
Breastw	**0.88**	0.61
Arrhythmia	0.76	**0.79**
Ionosphere	**0.75**	0.64
mean	*0.72*	*0.59*

Table 7.2: The AUC performance of SCiForest and iForest on data sets with added irrelevant attributes. Irrelevant attributes are added so that the total number of attributes is 512.

In this section, we have seen the advantage of SCiForest compared to iForest in handling data sets with added irrelevant attributes, without the help of an external attribute selector. However, there are data sets which do not respond well to SCiForest, e.g., Smtp, Pima and ForestCover. In the next section, we conduct two experiments to identify conditions in which SCiForest performs well with high dimensional data.

7.3 Understanding SCiForest's Detection Performance

In this section, we contrast two scenarios to understand a condition which helps SCiForest to correctly identify anomalies in high dimensional space. This condition is that when the anomalies are clustered, they are easier for SCiForest to detect because their numbers provide statistical evidence that some attributes are relevant when such evidence is lacking in the scattered anomalies. In the following experiments, we use iForest's result as a control measure. Ten-run averages are reported.

7.3.1 Scenario 1 (S1): Global clustered anomalies in high dimensional space

In S1, we first generate two data sets ($dsscl1$ and $dsscl2$), each containing a two-dimensional normal distribution of 2000 points as the normal points and 200 clustered anomalies. The covariance matrices used for generating these normal distributions are $(10, 3, 3, 2)$ and $(1, 2, 2, 8)$ respectively. Both normal distributions are centred at $(0, 0)$. The 200 clustered anomalies of both data sets are located at $(5, -5)$, outside of the normal distributions. These clustered anomalies are densely generated within a narrow range of 0.1. The anomalies in $dsscl1$ and $dsscl2$ are not correlated. The final data set is generated by combining $dsscl1$, $dsscl2$ and various irrelevant attributes column-wise into a single data set. Under this arrangement, there are anomalies which can only be identified by $dsscl1$ and anomalies that can only be identified by $dsscl2$. The ground truth of the final data set is the anomaly labels of $dsscl1$ and $dsscl2$. Figure 7.2 illustrates $dsscl1$ and $dsscl2$ with their clustered anomalies marked with a ($\triangle$) symbol.

From Section 5.2 in Chapter 5, we know that SCiForest is able to detect clustered anomalies and so we expect SCiForest to be able to detect the clustered anomalies in $dsscl1$ and $dsscl2$ before adding the irrelevant attributes. Interestingly, in Figure 7.3(a), we find that SCiForest is able to maintain an AUC close to 1 even after 50 irrelevant attributes have been added to $dsscl1$ and $dsscl2$. The reason why SCiForest is able to maintain such a high detection performance is that the $Sd_{reduction}$ criterion in SCiForest is sensitive to clustered anomalies, and as such hyper-planes with relevant attributes are selected when SCiForest's models are being trained. When the number of irrelevant attributes becomes large, the probability of selecting the relevant attributes in the trained model gradually decreases.

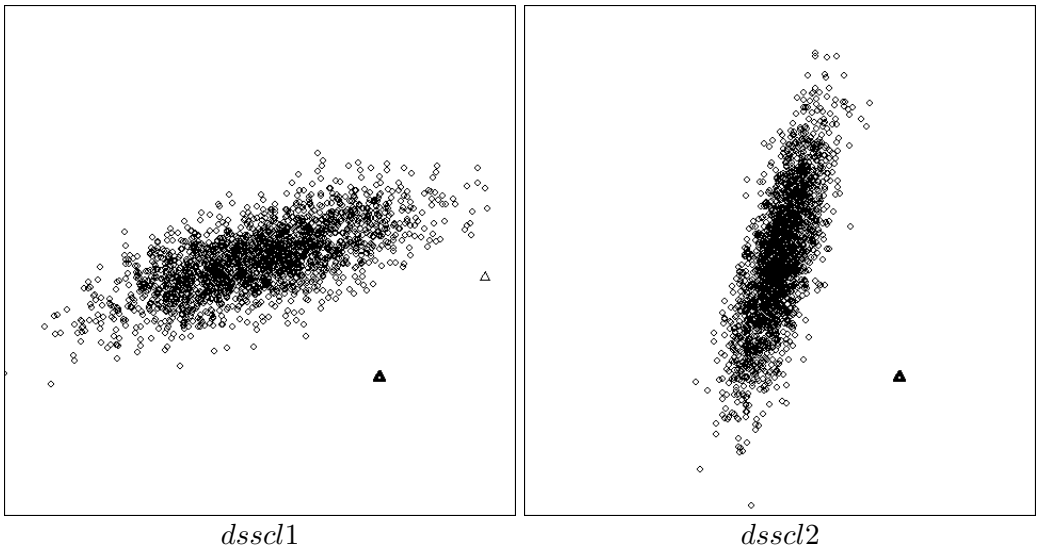
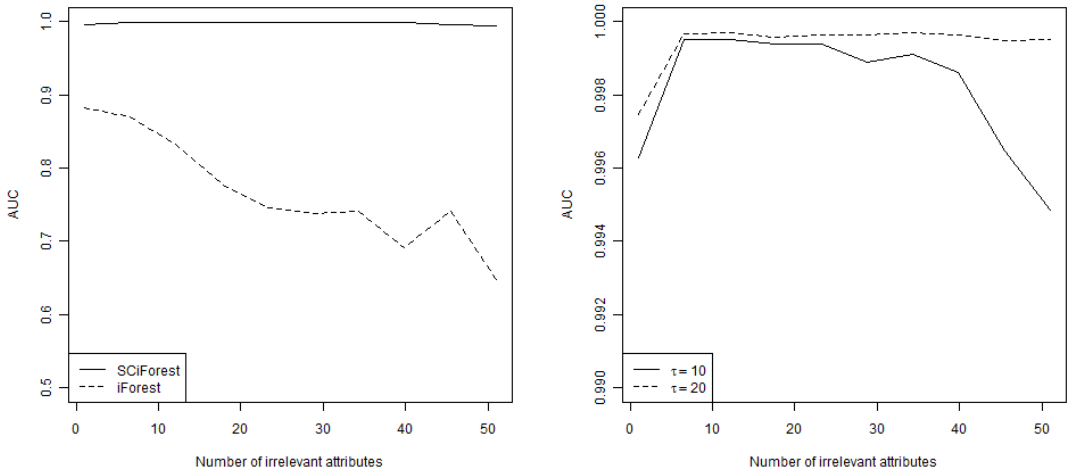


Figure 7.2: Scenario 1—distributions dsscl1 and dsscl2 each contains 200 clustered anomalies and 2000 normal points. Circle (○) denotes normal instances; triangle (△) denotes anomalies.

Since the parameter τ controls the number of hyper-planes considered in training the SCiForest models, in Figure 7.5(b), we find that setting $\tau = 20$ gives a better detection performance than $\tau = 10$ as $\tau = 20$ allows a higher probability of relevant attributes being selected in the trained model. Note that SCiForest's result in Figure 7.5(a) is the same curve ($\tau = 10$) in Figure 7.5(b); Figure 7.5(b) shows a very narrow range $(0.99 - 1)$ in the AUC.



(a) SCiForest vs. iForest (b) SCiForest with different τ

Figure 7.3: (a) Detection performance of SCiForest and iForest in S1, (b) Detection performance of SCiForest with different τ parameters. In data with a very high number of irrelevant attributes, SCiForest requires a greater τ parameter in order to maintain a high detection performance. Note that the AUC difference in (b) is very small.

7.3.2 Scenario 2 (S2): Local scattered anomalies in high dimensional space

In S2, we generate the same normal distributions as in S1 and name them $ds1$ and $ds2$. However, in S2, we instead assign the fringe points outside of the third quartile as anomalies. Figure 7.4 illustrates the two data sets $ds1$ and $ds2$. Similarly to S1, we generate irrelevant attributes with a uniform distribution. The anomalies in $ds1$ and $ds2$ are not correlated. The final data set is generated by combining $ds1$, $ds2$ and various irrelevant attributes column-wise into a single data set. The ground truth of the final data set is the anomaly labels of $ds1$ and $ds2$.

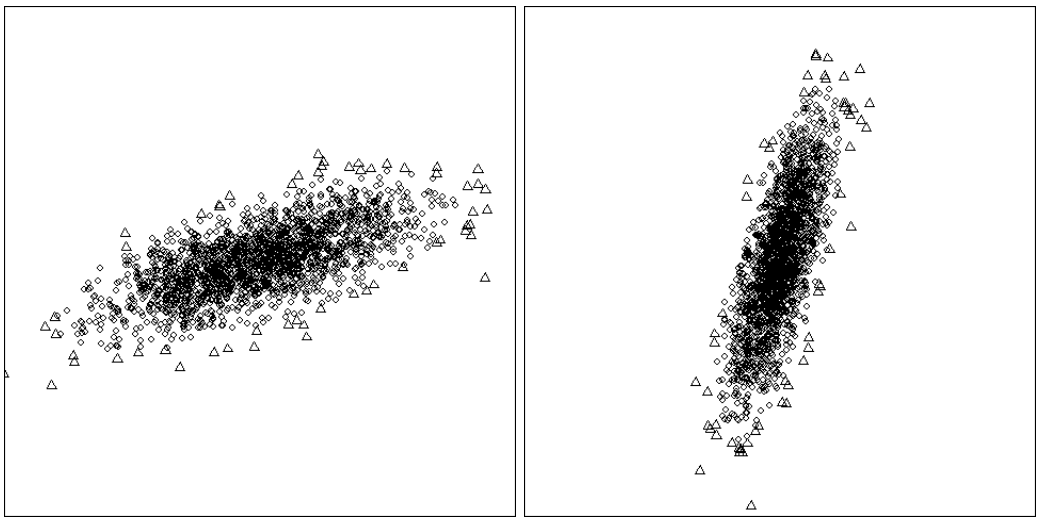


Figure 7.4: Scenario 2—the two two-dimensional distributions, $ds1$ and $ds2$ contain scattered anomalies. A circle (○) denotes the normal instances while a triangle (△) denotes the anomalies.

When comparing the detection performance of SCiForest and iForest in S2, we find that the detection performances of both detectors deteriorate when the number of irrelevant attributes increases. This deterioration is due to more and more irrelevant attributes being selected in building the models. Figure 7.5 compares the detection performance of SCiForest and iForest. Their performances are similar and SCiForest does not have a huge advantage over iForest in S2. The reason for this is that the $Sd_{reduction}$ criterion is unable to select the relevant attributes in S2.

To improve the performance of SCiForest and iForest in S2, it is possible to apply the kurtosis as an attribute selector, since the attributes relevant to the anomalies usually have a highly skewed distribution. In S2, the kurtosis values of the relevant attributes are

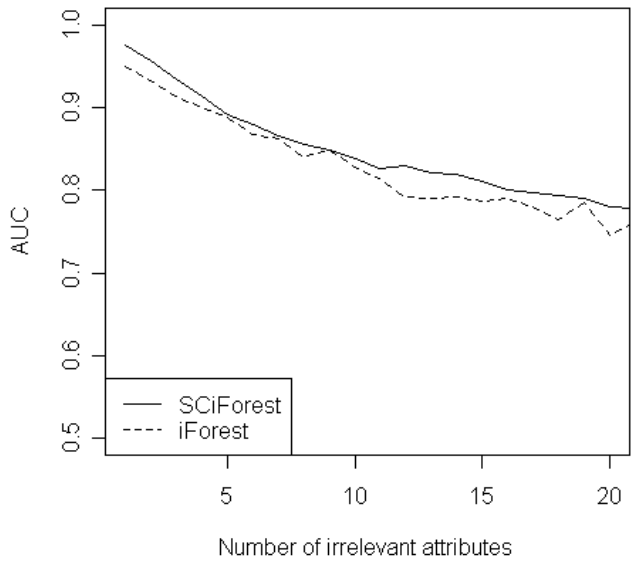


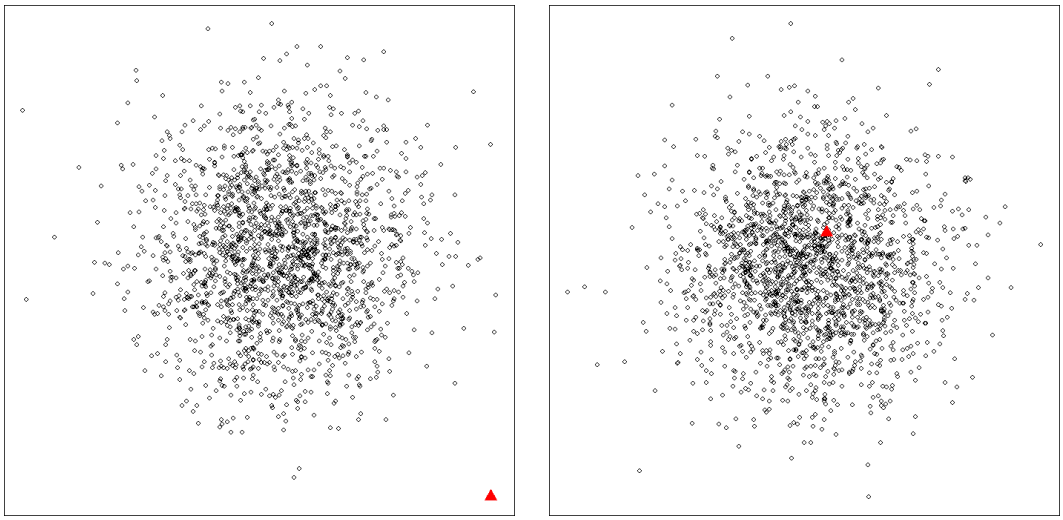
Figure 7.5: Scenario 2—the detection performance of SCiForest and iForest in S2.

always higher than those of the irrelevant attributes. The kurtosis values of the attributes in $ds1$ and $ds2$ are 0.106, 0.108, -0.144 and -0.167 while kurtosis values of the irrelevant attributes are less than -1.1. We can safely say that using kurtosis as an attribute selector would greatly improve the detection performance in S2. Although kurtosis is able to find the relevant attributes in a high dimensional setting, it is unclear whether we can use it to formulate a split selection criterion that replaces $Sd_{reduction}$. We will discuss this issue in Section 8.3 of Chapter 8.

In summary, as shown in Section 7.3.1, we find that SCiForest is able to determine the relevant attributes in high dimensional space with many irrelevant attributes as long as the anomalies are clustered. This condition is an important factor in SCiForest's better detection performance in high dimensional space. In contrast, in Section 7.3.2, our experimental result suggests that SCiForest performs poorly with scattered anomalies. This is because SCiForest is unable to determine the relevant attributes and its detection performance is reduced to a level similar to its randomized counterpart iForest. Aside from its original purpose in selecting good hyper-planes and split points in each node, we show that $Sd_{reduction}$ is also useful for selecting the relevant attributes in high dimensional space provided the anomalies are clustered.

7.4 Evaluation using a Synthetic Data Generator

Using the Mulcross data generator, we are able to generate high dimensional data containing clustered anomalies. The settings for Mulcross data generator are $a = 0.005, D = 20, cl = 1, n = 2000, d = \{1, ..., 300\}$, which produce 10 clustered anomalies in each data set with dimensionality ranging from 1 to 300. Note that the maximum dimensionality used in (Rocke and Woodruff, 1996) is 40. At the maximum dimensionality of 300, only a few attributes are able to distinguish the anomalies from normal instances as shown in Figure 7.6(a). Most other attributes depict the anomalies as points inside the cluster of the normal points (Figure 7.6(b)).



(a) (b)

Figure 7.6: Examples of data generated with Mulcross with 300 attributes. (a) shows the selected attributes that expose the anomalies (lower right corner); (b) illustrates the attributes in which the anomalies (inside the cluster) are indistinguishable from the normal instances. The black circles denote the normal instances and the red triangles denote the anomalies. The anomalies are very close together showing only one triangle in each diagram above.

In this experiment, we compare iForest, SCiForest and ORCA in an increasing number of dimensions. Both iForest and SCiForest use their default settings as stated in Table 4.3 of Chapter 4. For ORCA, we increase the k parameter to 20 so that k is large enough to detect the 10 clustered anomalies in this experiment.

Using all the dimensions available, Figure 7.7 reports the detection performance of the three detectors; it shows that SCiForest has the best detection performance followed by ORCA and iForest in the full range of dimensions. The AUC results presented are smoothed over ten-run averages. It takes about 120 dimensions for SCiForest's detection

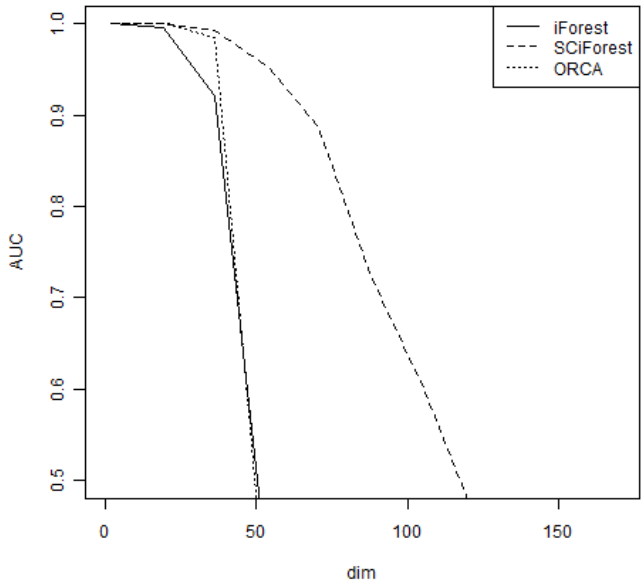


Figure 7.7: Using Mulcross data generator $a = 0.005, D = 20, cl = 1, n = 2000, d = \{1, ..., 300\}$, the above result shows that SCiForest has the best detection performance (AUC in x-axis) followed by ORCA and iForest, as the dimensionality increases (y-axis).

performance to fall through the 0.5 mark and 50 for iForest and ORCA. Since iForest employs a random attribute selection, iForest includes many irrelevant attributes in its models, causing a lower detection accuracy. ORCA suffers from the curse of dimensionality as the distance becomes increasingly meaningless in high dimensional space, while SCiForest is able to select the relevant attributes using $Sd_{reduction}$. This explains why SCiForest performs better than ORCA and iForest.

In this experiment, we show that SCiForest has a higher tolerance of high dimensional data compared to iForest and ORCA. Along with these high dimensional data sets, we also have experimented with random projections (Johnson and Lindenstrauss, 1984). We find that there is no obvious advantage to using random projections instead of the original attributes in these data sets. We conjecture that many of the attributes in the Mulcross data set are not useful in distinguish the anomalies from normal points and so random projections generated from these attributes are not beneficial for anomaly detection tasks.

7.5 Chapter Summary

This chapter looks at ways the isolation-based methods can deal with anomalies in high dimensional space. We find that by using a statistical test as an attribute selector, we are able to provide low cost dimensionality reduction for iForest. When kurtosis is applied on small sub-samples, iForest has the advantage of a faster processing time in large data sets. When kurtosis is applied on the entire training sample, the detection performance of iForest is generally better than when kurtosis is applied on small sub-samples.

We also find that SCiForest naturally selects relevant attributes through the use of its split selection criterion. SCiForest is effective when the anomalies are clustered, since the clustered anomalies allows SCiForest to determine the relevant attributes.

Using Mulcross to generate high dimensional data, we find that SCiForest is the least affected by the increase of dimensionality as compared to iForest and ORCA.

Chapter 8

Future Work

Isolation-based methods can be extended to deal with data streams (Section 8.1), categorical data (Section 8.2) and high dimensional data (Section 8.3).

8.1 Data streams

Data streams are potentially infinite, so an anomaly detector can only afford to look at the data once and must be able to cater for concept drift, forgetting outdated data and adapting to new data (Gama, 2010). To minimize the processing time, a trained model can be used throughout the entire data stream, provided it is updated when a change in concept occurs. Isolation-based methods can likely be applicable to on-line anomaly detection without major modifications. Since they only require a small subset of the training samples to train working models, they can start detecting early in a data stream. Their minimal training time also means that they can easily adapt to a concept change without much delay. We expect that this line of research will be very fruitful.

8.2 Categorical Data

Categorical data refers to data with discrete values without any ordering between these values and a finite number of possible values. Because of the lack of ordering, we cannot use categorical data directly in an isolation-based method. Also, with a finite (or usually very small) number of possible values, it implies that categorical data can be under-represented in the isolation-based framework due to fewer possible split points. Our goal is to handle both categorical and numerical attributes with the same isolation framework.

In Appendix E, we present a simple way to provide an ordering for categorical data in isolation-based methods. Our intuition is to (i) provide an ordering for categorical data and (ii) make categorical attributes compatible with numeric attributes in terms of being continuous. The rationale for (ii) is to not under represent categorical attributes as they will have as many possible split points as their numeric counterparts.

In the preliminary study in Appendix E, we find that a simple frequency ranking transformation is able to provide numerical orderings for categorical attributes so that they can be processed with isolation-based methods. We have shown a few examples of this transformation. However, this is not enough to confirm the effectiveness of this transformation. To do so, a more comprehensive study is required, which may involve other transformation methods and other anomaly detectors for comparison.

8.3 High Dimensional Data

Although we have presented several experimental results in Chapter 7 on different ways in which isolation-based methods can handle high dimensional data, the maximum number of dimensions we experiment with (512) is far less than the requirements of some modern data mining tasks, e.g., text mining. For isolation-based methods to handle an even higher number of dimensions, other possible extensions could explore different split-point selection criteria and different types of hyper-planes to isolate anomalies in high dimensional spaces. In terms of the split-point selection criteria, there are many different statistical tests that would be good candidates for selecting relevant attributes, e.g., Grubb's test (Livesey, 2007). More work is needed to formulate new split-point selection criteria that are able to select relevant attributes as well as good split points. Also, it may be possible to utilize a small number of labelled anomalies to guide the attribute selection process. However, this involves the cost of labelling the true anomalies. Also, so far we have only experimented with hyper-planes defined by simple linear equations. Hyper-planes defined by kernels would be an interesting research direction.

Chapter 9

Conclusions

This thesis proposes the first isolation-based anomaly detection method and makes four key contributions to the field of anomaly detection:

Firstly, we introduce the use of isolation as an effective and efficient means to detect anomalies compared to the commonly used distance and density measures. We show that the concept of isolation, when carried out in tree structures (or an isolation-framework), takes full advantage of the anomalies' properties of being 'few and different'. This isolation-framework isolates the anomalies closer to the root nodes while the normal points form long branches. This enables the average path length to be used as a measurement for the normality or abnormality. A short average path length indicates anomalies while a long average path length indicates normal points. We also demonstrate that anomaly definitions based on distance and density do not subsume the properties of being 'few and different' completely and show that isolation-based methods cover a wider range of anomalies, including scattered and clustered anomalies.

Secondly, we design isolation-based methods that have a low linear time complexity and very small memory requirement. In practice, we can build well performing isolation models with a small number of trees and small sub-samples of fixed size regardless of how large the data sets are. We demonstrate that a constant training time and space complexities can be achieved without compromising the predictive accuracy.

Thirdly, we introduce two ways to modify the isolation-based framework, in the use of the hyper-plane and deterministic split-point selection criterion, to increase its performance in detecting a very difficult class of anomalies—local clustered anomalies. This

gives rise to a variant called SCiForest. These modifications increase the ability to detect local clustered anomalies without a huge increase in the time and space complexities as compared to another variant, iForest, which uses an axis-parallel split and a random split-point selection criterion.

Fourth, our empirical comparison compares our isolation-based methods with three state-of-the-art anomaly detectors and shows that isolation-based methods are superior in terms of:

- their ability to handle clustered anomalies, especially in SCiForest,
- their runtime, predictive accuracies and memory requirements, especially for large data sets, and
- their robustness to a high number of anomalies.

In addition, we show that isolation-based methods are capable of being trained with or without anomalies. We also demonstrate their ability to deal with anomalies in high dimensional data by using an attribute selector and for SCiForest when the anomalies are clustered.

In future works, we expect isolation-based methods to be extended to handle data streams, categorical data and high dimensional data. Isolation-based methods are good candidates for detecting anomalies in data streams, since they are very computationally efficient as shown in this thesis. In terms of handling categorical data, a preliminary study is provided in Appendix E showing some early results in utilizing the same isolation-based framework to handle categorical data. This is made possible by a simple transformation process that converts the categorical values to numeric values. As for handling high-dimensional data, a further exploration of the split-point selection criteria and different forms of hyper-plane are expected to lead to improved methods.

References

Abe, N., Zadrozny, B. and Langford, J. (2006). Outlier detection by active learning, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, Philadelphia, PA, USA, pp. 504–509.

Achtert, E., Kriegel, H.-P., Reichert, L., Schubert, E., Wojdanowski, R. and Zimek, A. (2010). Visual evaluation of outlier detection models, *Database Systems for Advanced Applications, 15th International Conference, DASFAA 2010, Part II*.

Aggarwal, C. C. and Yu, P. S. (2001). Outlier detection for high dimensional data, *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, ACM Press, New York, NY, USA, pp. 37–46.

Aleskerov, E., Freisleben, B. and Rao, B. (1997). Cardwatch: a neural network based database mining system for credit card fraud detection, *Proceedings of IEEE Computational Intelligence for Financial Engineering* pp. 220–226.

Angiulli, F. and Fassetti, F. (2009). Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets, *ACM Transactions on Knowledge Discovery from Data (TKDD)* **3**(1): 1–57.

Angiulli, F. and Pizzuti, C. (2005). Outlier mining in large high-dimensional data sets, *IEEE Transactions on Knowledge and Data Engineering* **17**(2): 203–215.

Asuncion, A. and Newman, D. (2007). UCI machine learning repository.
URL: *http://www.ics.uci.edu/∼mlearn/MLRepository.html*

Banerjee, A., Chandola, V., Kumar, V., Srivastava, J. and Lazarevic, A. (2008). Anomaly detection: A tutorial, 2008 SIAM Conference on Data Mining.

Barnett, V. and Lewis, T. (1994). *Outliers in statistical data*, 3rd edn, John Wiley & Sons, Chichester.

Bay, S. D. and Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule, *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, Washington, D.C., pp. 29–38.

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbor" meaningful?, *ICDT '99: Proceedings of the 7th International Conference on Database Theory*, Springer-Verlag, London, UK, pp. 217–235.

Bishop, C. M. (1994). Novelty detection and neural network validation, *Vision, Image and Signal Processing, IEE Proceedings-* **141**(4): 217–222.

Blackard, J. A. (1998). *Comparison of neural networks and discriminant analysis in predicting forest cover types*, PhD thesis, Fort Collins, CO, USA. AAI9921979.

Boriah, S., Chandola, V. and Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation, *SDM*, SIAM, pp. 243–254.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). LOF: identifying density-based local outliers, *ACM SIGMOD Record* **29**(2): 93–104.

Caputo, B., Sim, K., Furesjo, F. and Smola, A. (2002). Appearance-based object recognition using SVMs: which kernel should I use?, *Proc of NIPS workshop on Statistical methods for computational experiments in visual processing and computer vision*, Whistler.

Chandola, V., Banerjee, A. and Kumar, V. (2009). Anomaly detection: A survey, *ACM Computing Surveys* .

Chaudhary, A., Szalay, A. S., Szalay, E. S. and Moore, A. W. (2002). Very fast outlier detection in large multidimensional data sets, *The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, ACM Press, Madison, Wisconsin.

Chawla, N. V., Japkowicz, N. and Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets, *SIGKDD Explor. Newsl.* **6**(1): 1–6.

Chen, Y., Dang, X., Peng, H. and Bart Jr., H. L. (2009). Outlier detection with the kernelized spatial depth function, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**: 288–305.

Dasgupta, D. and Majumdar, N. S. (2002). Anomaly detection in multidimensional data using negative selection algorithm, *CEC '02: Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress*, IEEE Computer Society, Washington, DC, USA, pp. 1039–1044.

Dasgupta, D. and Nino, F. (2000). A comparison of negative and positive selection algorithms in novel pattern detection, Vol. 1, pp. 125 –130 vol.1.

D'haeseleer, P., Forrest, S. and Helman, P. (1996). An immunological approach to change detection: Algorithms, analysis and implications, *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, IEEE Computer Society, Washington, DC, USA, p. 110.

Drapper, N. R. and Smith, H. (1966). *Applied Regression Analysis*, Wiley.

Dutta, H., Giannella, C., Borne, K. D. and Kargupta, H. (2007). Distributed top-k outlier detection from astronomy catalogs using the demac system., *Proceedings of the Seventh SIAM International Conference on Data Mining*, SIAM.

Ertoz, L., Eilertson, E., Lazarevic, A., P.-N. Tan, V. K., Srivastava, J. and Dokas, P. (2004). *Data Mining - Next Generation Callenges and Future Directions*, MIT, chapter MINDS - Minnesota Intrusion Detection System.

Fan, H., Zaïane, O. R., Foss, A. and Wu, J. (2006). A nonparametric outlier detection for effectively discovering top-n outliers from engineering data., *Proceedings the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2006*, pp. 557–566.

Fan, W., Miller, M., Stolfo, S. J., Lee, W. and Chan, P. K. (2001). Using artificial anomalies to detect unknown and known network intrusions, *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, DC, USA, pp. 123–130.

Fawcett, T. (2006). An introduction to roc analysis, *Pattern Recogn. Lett.* **27**: 861–874. **URL:** *http://portal.acm.org/citation.cfm?id=1159473.1159475*

Fawcett, T. and Provost, F. (1997). Adaptive fraud detection, *Data Mining and Knowledge Discovery* **1**(3): 291–316.

Filzmoser, P., Maronna, R. and Werner, M. (2008). Outlier identification in high dimensions, *Computational Statistics and Data Analysis* **52**(3): 1694–1711.

Gama, J. (2010). *Knowledge Discovery from Data Streams*, Chapman & Hall/CRC.

Ghoting, A., Otey, M. E. and Parthasarathy, S. (2004). Loaded: Link-based outlier and anomaly detection in evolving data sets, *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, DC, USA, pp. 387–390.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K. and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, *Circulation* **101**(23): e215–e220.

Hand, D. J. (2009). Measuring classifier performance: a coherent alternative to the area under the roc curve, *Machine Learning* **77**(1): 103–123.

Hawkins, D. M. (1980). *Identification of Outliers*, Chapman and Hall, London; New York.

Hawkins, D. M., Bradu, D. and Kass, G. V. (1984). Location of several outliers in multiple-regression data using elemental sets, *Technometrics* **26**(3): 197208.

He, Z., Deng, S. and Xu, X. (2005a). An optimization model for outlier detection in categorical data, *in* D.-S. Huang, X.-P. Zhang and G.-B. Huang (eds), *ICIC (1)*, Vol. 3644 of *Lecture Notes in Computer Science*, Springer, pp. 400–409.

He, Z., Deng, S. and Xu, X. (2005b). A unified subspace outlier ensemble framework for outlier detection., *in* W. Fan, Z. Wu and J. Yang (eds), *WAIM*, Vol. 3739 of *Lecture Notes in Computer Science*, Springer, pp. 632–637.

He, Z., Xu, X. and Deng, S. (2003). Discovering cluster-based local outliers, *Pattern Recognition Letters* **24**(9-10): 1641–1650.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies, *Artifical Intelligence Review* **22**(2): 85–126.

Janeja, V. P. and Atluri, V. (2008). Random walks to identify anomalous free-form spatial scan windows, *IEEE Transactions on Knowledge and Data Engineering* **20**(10): 1378–1392.

Joanes, D. N. and Gill, C. A. (1998). Comparing measures of sample skewness and kurtosis, *Journal of the Royal Statistical Society (Series D): The Statistician* **47**(1): 183–189.

Johnson, W. and Lindenstrauss, J. (1984). Extensions of lipschitz mappings into a hilbert space, *Contemporary Mathematics* **26**: 189–206.

Joshi, M. V., Agarwal, R. C. and Kumar, V. (2001). Mining needle in a haystack: classifying rare classes via two-phase rule induction, *SIGMOD Record* **30**(2): 91–102.

Joshi, M. V., Agarwal, R. C. and Kumar, V. (2002). Predicting rare classes: can boosting make any weak learner strong?, *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, New York, NY, USA, pp. 297–306.

Knorr, E. M. and Ng, R. T. (1998). Algorithms for mining distance-based outliers in large datasets, *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*, Morgan Kaufmann, San Francisco, CA, USA, pp. 392–403.

Knorr, E. M., Ng, R. T. and Tucakov, V. (2000). Distance-based outliers: algorithms and applications, *The VLDB Journal* **8**(3-4): 237–253.

Knuth, D. E. (1968). *The art of computer programming*, Addison-Wiley.

Knuth, D. E. (1998). *Art of Computer Programming, Volume 3: Sorting and Searching (2nd Edition)*, Addison-Wesley Professional.

Knuth, D. E. (2006). *Art of Computer Programming, Volume 4, Fascicle 4: Generating All Trees*, Addison-Wesley Professional.

Lazarevic, A. and Kumar, V. (2005). Feature bagging for outlier detection, *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ACM, New York, NY, USA, pp. 157–166.

Liu, R. Y., Parelius, J. M. and Singh, K. (1999). Multivariate analysis by data depth: Descriptive statistics, graphics and inference, *The Annals of Statistics* **27**(3): 783–840.

Livesey, J. H. (2007). Kurtosis provides a good omnibus test for outliers in small samples, *Clinical Biochemistry* **40**(13-14): 1032–1036.

Moonesignhe, H. D. K. and Tan, P.-N. (2006). Outlier detection using random walks, *ICTAI '06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, IEEE Computer Society, Washington, DC, USA, pp. 532–539.

Murphy, R. B. (1951). *On Tests for Outlying Observations*, PhD thesis, Princeton University.

Papadimitriou, S., Kitagawa, H., Gibbons, P. and Faloutsos, C. (2003). Loci: fast outlier detection using the local correlation integral, *Data Engineering, 2003. Proceedings. 19th International Conference on* pp. 315–326.

Pearson, R. K. (2005). *Mining Imperfect Data: Dealing with Contamination and Incomplete Records*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Phua, C., Alahakoon, D. and Lee, V. (2004). Minority report in fraud detection: classification of skewed data, *SIGKDD Explorations Newsletter* **6**(1): 50–59.

Preiss, B. R. (1999). *Data Structures and Algorithms with Object-Oriented Design Patterns in Java*, Wiley.

Ramaswamy, S., Rastogi, R. and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets, *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, ACM Press, New York, NY, USA, pp. 427–438.

Rocke, D. M. and Woodruff, D. L. (1996). Identification of outliers in multivariate data, *Journal of the American Statistical Association* **91**(435): 1047–1061.

Rousseeuw, P. J. (1984). Least median of squares regression, *Journal of the American Statistical Association* **79**: 871–880.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust regression and outlier detection*, John Wiley & Sons, Inc., New York, NY, USA.

Ruskey, F. (1980). On the average shape of binary trees, *SIAM Journal on Algebraic and Discrete Methods* **1**(1): 43–50.

Schilling, M. F., Watkins, A. E. and Watkins, W. (2002). Is human height bimodal?, *The American Statistician* **56**: 223–229.

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J. and Platt, J. (2000). Support vector method for novelty detection, Vol. 12, pp. 582–588.

Shi, T. and Horvath, S. (2006). Unsupervised learning with random forest predictors, *Journal of Computational and Graphical Statistics* **15**(1): 118–138.

Snedecor and Cochran (1989). *Statistical Methods*, Iowa State University Press.

Song, X., Wu, M., Jermaine, C. and Ranka, S. (2007). Conditional anomaly detection, *IEEE Transactions on Knowledge and Data Engineering* **19**(5): 631–645.

Steinwart, I., Hush, D. and Scovel, C. (2005). A classification framework for anomaly detection, *Journal of Machine Learning Research* **6**: 211–232.

Sun, J., Xie, Y., Zhang, H. and Faloutsos, C. (2007). Less is more: Compact matrix decomposition for large sparse graphs, *Proceedings of the Seventh SIAM International Conference on Data Mining (SDM 2007)*.

Sun, P. (2006). *Outlier detection in high dimensional, spatial and sequential data sets*, PhD thesis, The University of Sydney.

Tan, P.-N., Steinbach, M. and Kumar, V. (2005). *Introduction to Data Mining*, Addison-Wesley.

Tang, J., Chen, Z., Fu, A. W.-C. and Cheung, D. W.-L. (2002). Enhancing effectiveness of outlier detections for low density patterns, *PAKDD '02: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Springer-Verlag, London, UK, pp. 535–548.

Tao, Y., Xiao, X. and Zhou, S. (2006). Mining distance-based outliers from large databases in any metric space, *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, New York, NY, USA, pp. 394–403.

Tax, D. M. J. and Duin, R. P. W. (2004). Support vector data description, *Machine Learning* **54**(1): 45–66.

Tukey, J. W. (1977). *Exploratory Data Analysis*, Addison-Wesley.

Vilalta, R. and Ma, S. (2002). Predicting rare events in temporal domains, *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, DC, USA, p. 474.

Wettschereck, D. (1994). *A study of distance-based machine learning algorithms*, PhD thesis, Oregon State University. Adviser-Thomas G. Dietterich.

Williams, G., Baxter, R., He, H., Hawkins, S. and Gu, L. (2002). A comparative study of rnn for outlier detection in data mining, *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, DC, USA, p. 709.

Wong, W.-K., Moore, A., Cooper, G. and Wagner, M. (2003). Bayesian network anomaly pattern detection for disease outbreaks, *in* T. Fawcett and N. Mishra (eds), *Proceedings of the Twentieth International Conference on Machine Learning*, AAAI Press, Menlo Park, California, pp. 808–815.

Yamanishi, K., Takeuchi, J.-I., Williams, G. and Milne, P. (2000). On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, pp. 320–324.

Yu, X., Tang, L. A. and Han, J. (2009). Filtering and refinement: A two-stage approach for efficient and effective anomaly detection, *IEEE International Conference on Data Mining* pp. 617–626.

Zhang, Y., Meratnia, N. and Havinga, P. J. M. (2007). A taxonomy framework for unsupervised outlier detection techniques for multi-type data sets, *Technical Report TR-CTIT-07-79*, Enschede.

Appendix A

Technical Details for The Empirical Evaluation

A.1 Twelve Natural Data Sets

The twelve natural data sets used for evaluation can be found in UCI repository[1] (Asuncion and Newman, 2007).

The description of the twelve natural data sets are as follows:

- The *http* and *smtp* data sets are taken from (Yamanishi et al., 2000). The original KDD CUP 99 network intrusion data set has 41 attributes (34 continuous and 7 categorical) and class labels. The class labels are made up of 22 attack types and the 'normal' operation. The original data set contains 80.1% of attacks, 3,925,651 out of 4,898,431. To make it possible to apply the anomaly detection methods, a subset of 976,157 records, including 3,377 attacks (0.35%) is selected. Four attributes {service, duration, src_bytes, dst_bytes} out of the 41 attributes are selected as they were recognized as the most basic attributes (Yamanishi et al., 2000). The data are then further divided into five subsets {http, smtp, ftp, ftp_data, others} according to the labels in the service attribute. The *http* and *smtp* are the two largest subsets. The remaining three attributes {duration, src_bytes, dst_bytes} are transformed by $y = log(x + 0.1)$ so that they are not concentrated around 0.

- The *Annthyroid* data set was used in anomaly detection, e.g. (Abe et al., 2006). In this thesis, classes 1 and 2 both are labelled as anomalies rather than using the

[1] `http://archive.ics.uci.edu/ml/`

same data set twice with the different class being the labelled anomalies. In the original problem, the class 1 refers to 'increased binding protein' meaning hyper-functioning thyroid, the class 2 'decreased binding protein' meaning subnormal-functioning thyroid and only the class 3 'negative' is normal.

- The *Arrhythmia* data set was used in (He et al., 2005b). As used in the previous work, classes 03, 04, 05, 07, 08, 09, 14 and 15 are regarded as the anomalies, which are less than 5% of the data.

- The *Dermatology* data set is converted into an anomaly detection problem by labelling class 6 'pityriasis rubra pilaris' (PRP), a rare condition, the smallest class, as the anomalies. For this data set, the attribute 34 'Age' is removed so that the anomalies are detected regardless of age.

- The *ForestCover* data set is being converted to an anomaly detection problem by retaining the largest and the smallest classes (classes 2 'Lodgepole Pine' and 4 'Cottonwood/Willow'). The smallest class 'Cottonwood/Willow' is being labelled as the anomalies. This data set was used in (Bay and Schwabacher, 2003).

- The *Ionosphere* data set is also an anomaly detection problem by labelling class 'bad' as anomalies. The class 'bad' is transmitted radar signals pass through the ionosphere without returning. The class 'good' are signals returns that show some type of structures in the ionosphere.

- The *Pima* data set is converted to an anomaly detection problem by labelling the positive diabetics as the anomalies. In this data set, anomaly detection can be seen as detecting the high risk patients which are more likely to be affected by diabetes.

- The *Satellite* or Landsat Multi-Spectral Scanner Image data set has 6 classes. This data set is converted to an anomaly detection problem by labelling the three smallest classes, 'damp grey soil', 'vegetation stubble' and 'cotton crop' as the anomalies. This data was used in (Lazarevic and Kumar, 2005) with only the smallest class labelled as anomalies, however, the observed AUC is very close to 0.5 for many detectors, which is meaningless in anomaly detection.

- The *Shuttle* data set is used in (Lazarevic and Kumar, 2005). In this thesis, the anomalies are the classes 2,3,5,6 and 7 and the normal points are the class 1.

- The *Breastw* or Wisconsin Breast Cancer data set was used in (Williams et al., 2002). The malignant instances are labelled as the anomalies.

- The *Mammography* was used in (Lazarevic and Kumar, 2005). Class 2 is labelled as the anomalies.

A.2 Synthetic Data Generator

The Mulcross data generator[2] is designed to evaluate anomaly detectors. Mulcross generates a multi-variate normal distribution with a selectable number of anomaly clusters. The Mulcross data generator generates anomalies that have a covariance matrix same as that of the normal instances. In other words, the normal instances and the anomaly clusters have the same shape. In the context of the statistical-based outlier detection, it is suggested that if this kind of anomalies can be detected, then other kinds of anomalies should be detected as well (Rocke and Woodruff, 1996). This kind of anomalies is also known as *shift outliers* (Hawkins, 1980) in which the normal instances and the anomalies are drawn from the same multivariate normal distribution with the anomalies displaced. With Mulcross, users can adjust the number of abnormal clusters cl, the number of points generated n, the ratio of "bad" data a, the distance between the anomalies and the normal points D, and the number of dimensions d. Throughout this thesis, the same symbols will be used to denote the settings of Mulcross. An example of Mulcross generated data can be found in Figure A.1.

A.3 Isolation Forest package in R

I provide a publicly available implementation of iForest and it can be found in:
`http://sourceforge.net/projects/iforest/`.
It is intended to be used with R[3] as an add-on library. For the windows operating system, after downloading the file `IsolationForest_0.0-25.zip`. In an R console, to install the implementation, go to `menu->Packages->Install package(s) from local zip files...` and select the downloaded file. You only have to do this once. To load the package in R, type in `library(IsolationForest)`. If the package is correctly loaded, the

[2] `http://lib.stat.cmu.edu/jasasoftware/rocke`
[3] `http://www.r-project.org/`

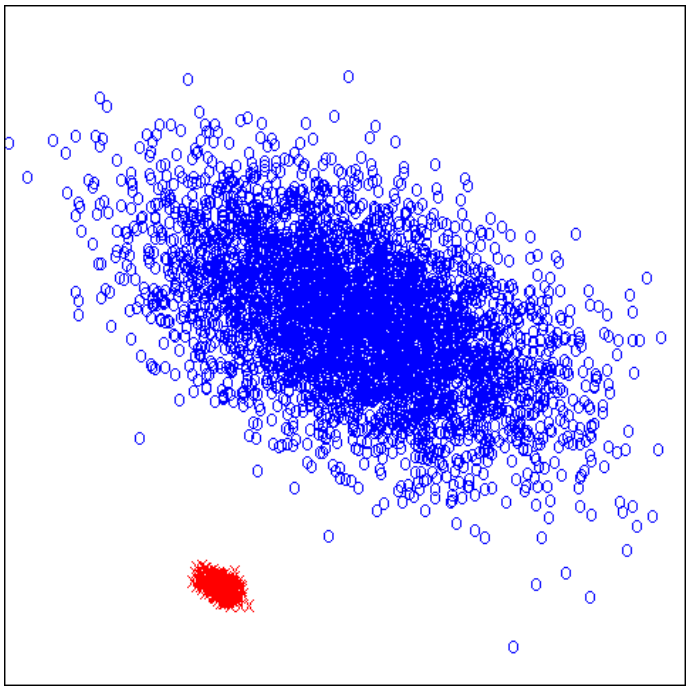


Figure A.1: An example of data generated by the Mulcross data generator.

	n	d	anomaly class
Http	567497	3	attack (0.4%)
ForestCover	286048	10	class 4 (0.9%) vs. class 2
Smtp	95156	3	attack (0.03%)
Shuttle	49097	9	classes 2,3,5,6,7 (7%)
Mammography	11183	6	class 1 (2%)
Annthyroid	6832	6	classes 1, 2 (7%)
Satellite	6435	36	3 smallest classes (32%)
Pima	768	8	pos (35%)
Breastw	683	9	malignant (35%)
Arrhythmia	452	274	classes: 03,04,05,07, 08,09,14,15 (15%)
Dermatology	366	33	largest class vs. smallest class (5%)
Ionosphere	351	32	bad (36%)

Table A.1: Properties of the data used for empirical evaluation, where n is the number of instances, and d is the number of dimensions, and the percentage in bracket indicates the percentage of anomalies.

R console displays the name and the version number of this package. The main functions in this package are the `IsolationTrees` and `AnomalyScore` functions. Information about these functions is available by typing: `?IsolationTrees` at the R console. At the time of writing, this package only includes the implementation of iForest. The implementation of SCiForest will be included in the package at a later stage. For the `IsolationTrees`

function, the default setting of iForest used in this thesis is: `ntree=100, rFactor=1, rowSamp=T, nRowSamp=256`.

A.4 Implementations of Other Anomaly Detectors Used

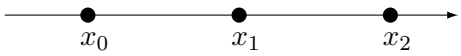
In this thesis, three other anomalies detectors are used for demonstration and empirical evaluation. They are the ORCA (Bay and Schwabacher, 2003), LOF (Knorr and Ng, 1998) and SVM (Schölkopf et al., 2000). Respectively, they represent the state-of-the-art distance-, density- and model-based anomaly detectors. For this thesis, the implementation of ORCA can be found in: `http://www.stephenbay.net/orca/`. The implementation of LOF can be found in the R package `dprep`, under the function `lofactor(data,k)`. This implementation can be slower than other LOF implementations, due to the use of scripting language. However, the `dprep` package is well tested and it is available since 2005. The implementation of SVM can be found in the R package `kernlab`, under the function `ksvm(...)`. The parameter setting for `ksvm` is: `type="one-svc", kernel = "rbfdot", kpar = "automatic"`. This setting performs the novelty detection using the Radial Basis kernel (Gaussian) with the selection of hyper-parameters, including *sigma* inverse kernel width, using the heuristics in the function `sigest(...)` (Caputo et al., 2002). More details on these implementations can be found in their respective web sites and help pages.

Appendix B

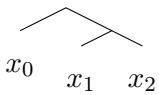
A Probabilistic Explanation to Isolation Trees

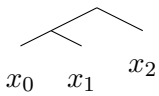
In order to understand the susceptibility of Isolation in terms of the average path length, we analyse the property of the expected path length in relation to the underlying data distribution. For univariate cases, the expected path length of every point in a distribution can be summarized as a summation series from all the possible isolation trees. For symbols and notations, please refer to Table B.1.

Let us start with a simple model. Given a sample of univariate, $\{x_0, x_1, x_2\} \in X$, where $x_0 < x_1 < x_2$, as shown below.



There are two possible tree structures that can isolate these points:

(Tree A) 

(Tree B) 

If the initial split point falls between x_0 and x_1, then Tree A is constructed, else Tree B. The expected path length for each point can be calculated exactly according to the possible tree structures mentioned above:

$$E(h(x_0)) = P(h(x_0) = 1) \times 1 + P(h(x_0) = 2) \times 2,$$

$$E(h(x_1)) = P(h(x_1) = 1) \times 1 + P(h(x_1) = 2) \times 2,$$

$$E(h(x_2)) = P(h(x_2) = 1) \times 1 + P(h(x_2) = 2) \times 2.$$

$\mathbf{x}$	a data point, $\mathbf{x} = [x^1, ..., x^d]$
X	a data set $\mathbf{x} \in X$
n	number of data points in a data set, $n = \|X\|$
m	index of data point $\mathbf{x}_m$, $m \in \{0, ..., n-1\}$
Q	a set of attributes
d	number of attributes, $d = \|Q\|$
r	an attribute
T	a tree or a node
t	number of trees
h	path length or function $h(\mathbf{x})$ which returns the path length of $\mathbf{x}$
$hlim$	evaluation height limit
ψ	sub-sampling size
l	a possible path length
cl	number of anomaly clusters
$P(.)$	The probability function
$E(.)$	The expectation function
$c(n)$	average height of random binary trees,$c(n) = 2H(n-1) - 2(n-1)/n$
$H(i)$	harmonic number $H(i) \approx \ln(i) + 0.5772156649$ (Euler's constant)
C_p	The Catalan Number, $C_p = \binom{2p}{p} - \binom{2p}{p-1}$
C_{pr}	The Catalan Number, $C_{pr} = \binom{p+r}{p} - \binom{p+r}{p-1}$

Table B.1: Symbols and Notations

Let $D_{i,j}$ be the distance between x_i and x_j. In this simple example, the first split point of a tree basically decides which possible structure will be formed. Assuming uniform distribution, the probability of $P(h(x_0) = 1) = P(h(x_2) = 2) = P(A) = \frac{D_{0,1}}{D_{0,2}}$ and $P(h(x_0) = 2) = P(h(x_2) = 1) = P(B) = \frac{D_{1,2}}{D_{0,2}}$.

In general, where $|X| > 1$, the number of possible trees is C_j, a Catalan number of j, where $j = |X| - 1$. C_j grows as follows: $\{1, 2, 5, 14, 42, 132, 429, 1430, 4862\}$ for $|X| \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Note that in cases where the number of samples is more than three, each probabilistic component is mapped to multiple possible tree structures. For each data point of interest x, the expected path length $E(h(x))$ is a summation of a series of possible path lengths with probabilistic components such that:

$$E(h(x)) = \sum_{l=1}^{|X|-1} P(h(x) = l) \times l. \tag{B.1}$$

The possible path lengths for any fringe points are $h(x) \in [1, |X| - 1]$; and the possible path lengths for any non-fringe points are $h(x) \in [2, |X| - 1]$. The sum of the probabilistic components of all the possible path lengths $\sum_l P(h(x) = l) = 1$.

This analysis using all possible trees has been conducted by (Knuth, 2006; Ruskey, 1980).

Assuming that each possible tree structure is equally probable, i.e., under uniform distribution; the term $P(h(x) = l)$ can be estimated by $\frac{t_{lmj}}{C_j}$(Knuth, 2006), where t_{lmj} is the total number of possible trees that has $h(x_m) = l$ with j number of internal nodes,

$$t_{lmj} = \sum_{u=0}^{n} \begin{pmatrix} l \\ u \end{pmatrix} C_{(m-u)(m-l)} C_{(j-m-l+u)(n-m-l)}. \tag{B.2}$$

The average path length of a data point x_m is $E(h(x_m)) = \frac{h_{mj}}{C_j}$(Knuth, 2006), where h_{mj} is the sum of path lengths for x_m in all the possible trees with j number of internal nodes,

$$h_{mj} = 2 \begin{pmatrix} 2m \\ m \end{pmatrix} \begin{pmatrix} 2j-2m \\ j-m \end{pmatrix} \frac{(2m+1)(2j-2m+1)}{(j+1)(j+2)} - C_j, \tag{B.3}$$

and $m \in \{0, ..., j\}$.

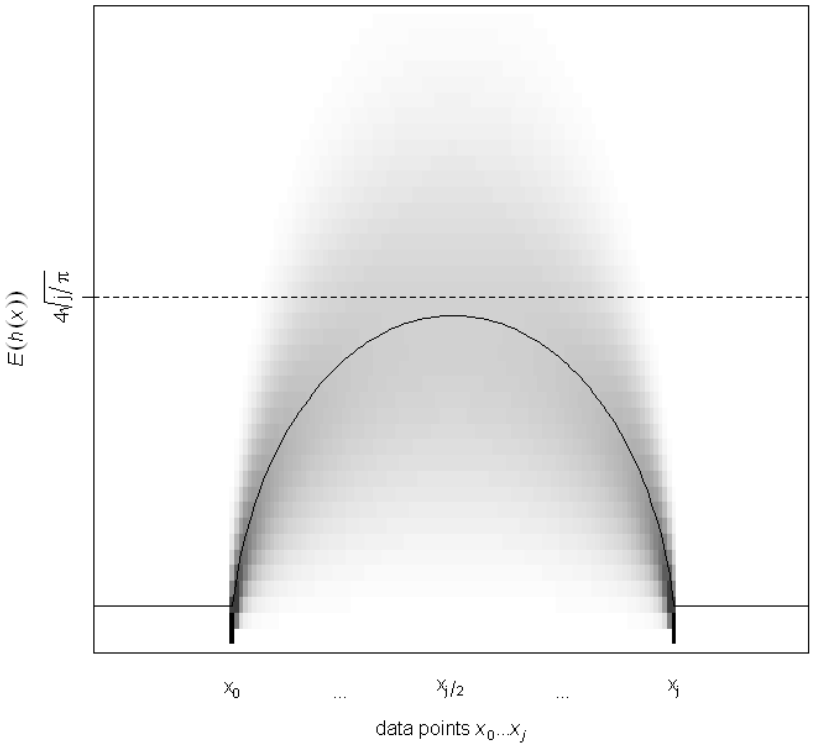


Figure B.1: The average path length $E(h(x))$ of randomly generated binary trees for evenly spaced data points. $P(h(x))$ is represented by grey scale distribution.

The average path length of all the possible trees[1] was initially investigated in (Ruskey, 1980). It is shown in Figure B.1, which has a **dome** shape. The probabilistic component $P(h(x) = l)$ in Equation B.1 gives the grey scale distribution in Figure B.1. The height of this shape is approximately $4\sqrt{j/\pi}$. For regions that have no data point, i.e., $x < x_0$ and $x > x_j$, those regions share the same path length as their closest fringe points. In Figure B.1, the dome shape reveals that points on either sides of a distribution (fringe points) have much lower expected path lengths compared to those inside the main distribution (core points).

[1]Ruskey called this the shape of the random trees.

Appendix C

Detail Empirical Results

C.1 iForest with various number of trees (t)

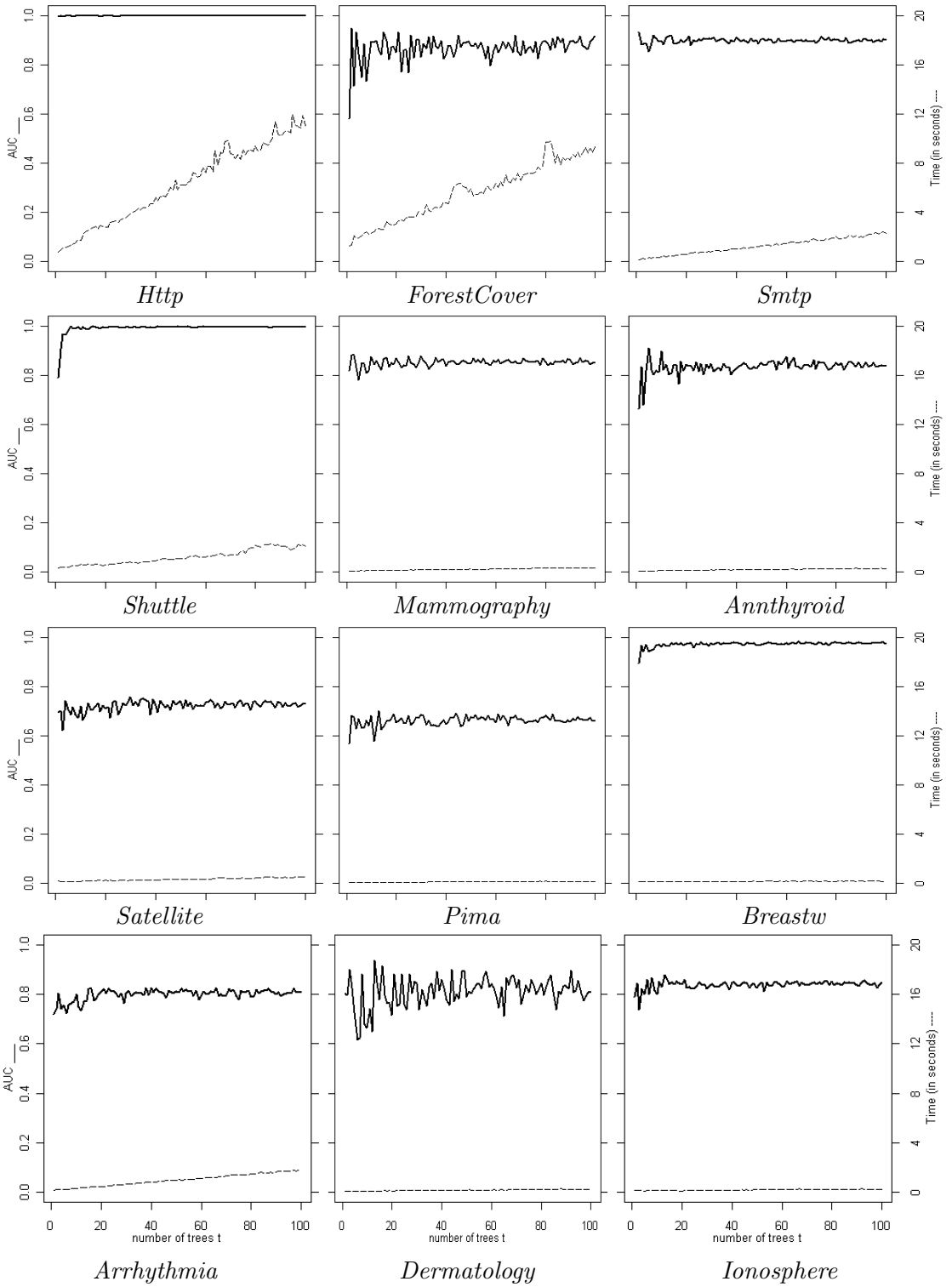


Figure C.1: iForest's AUC performance (y1-axis) and processing time (y2-axis) versus the different number of trees t (x-axis). The sub-sampling size ψ used is 256.

C.2 iForest with various sub-sampling sizes (ψ)

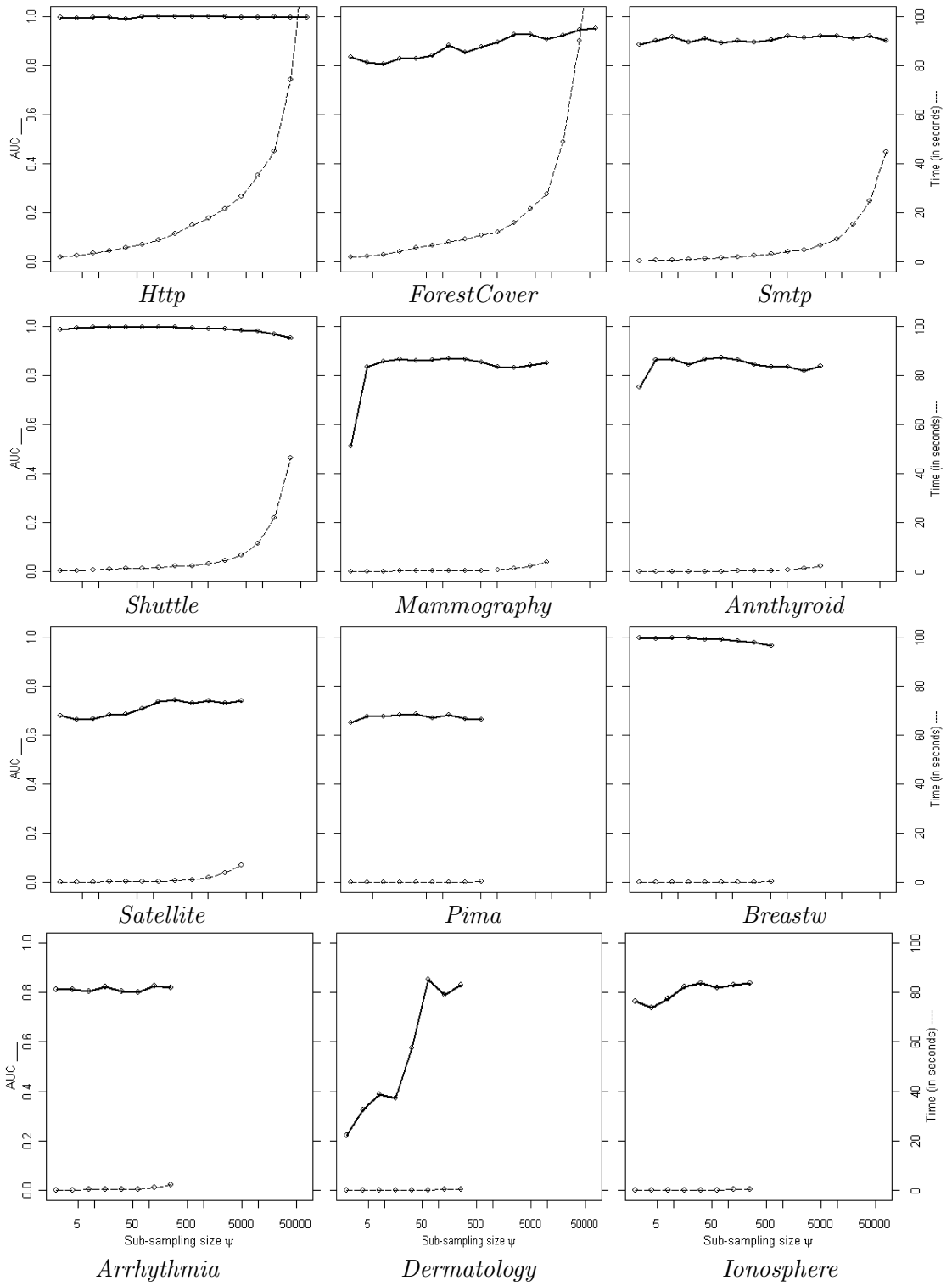


Figure C.2: iForest's AUC performance (y1-axis) and processing time (y2-axis) versus the different sub-sampling size ψ settings in log scale (x-axis). The number of trees t in each model is 100.

C.3 iForest's performance using Kurtosis as feature selector

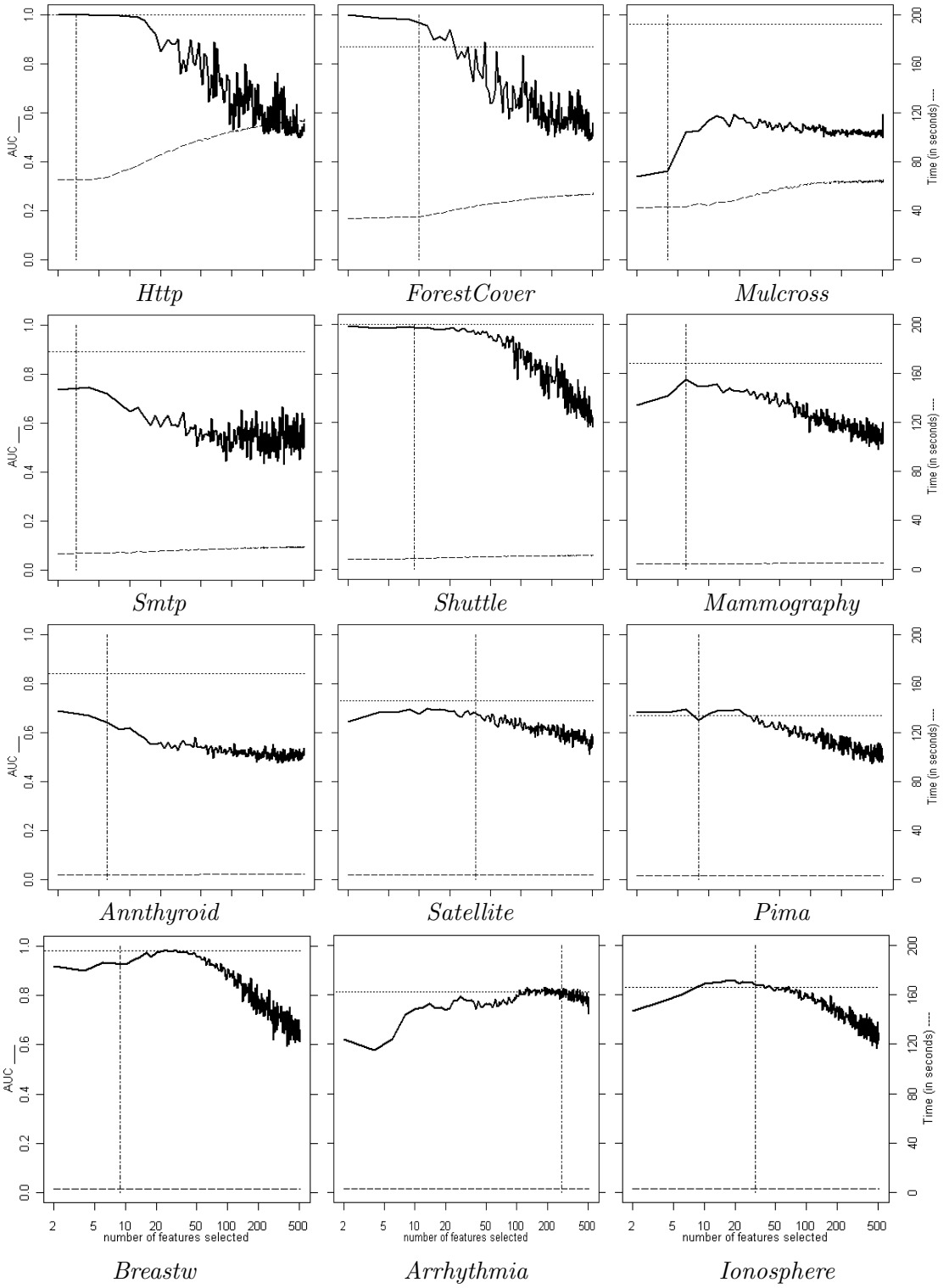


Figure C.3: iForest's AUC performance (y1-axis, solid lines) and processing time (y2-axis, dashed lines) versus the different number of attributes selected from Kurtosis (x-axis, log scale) in irrelevant attribute added data. Dotted lines denote the AUC performance of iForest using the original data without added irrelevant attributes. Dotted dash lines indicate the original number of attributes.

C.4 SCiForest with various number of trees (t)

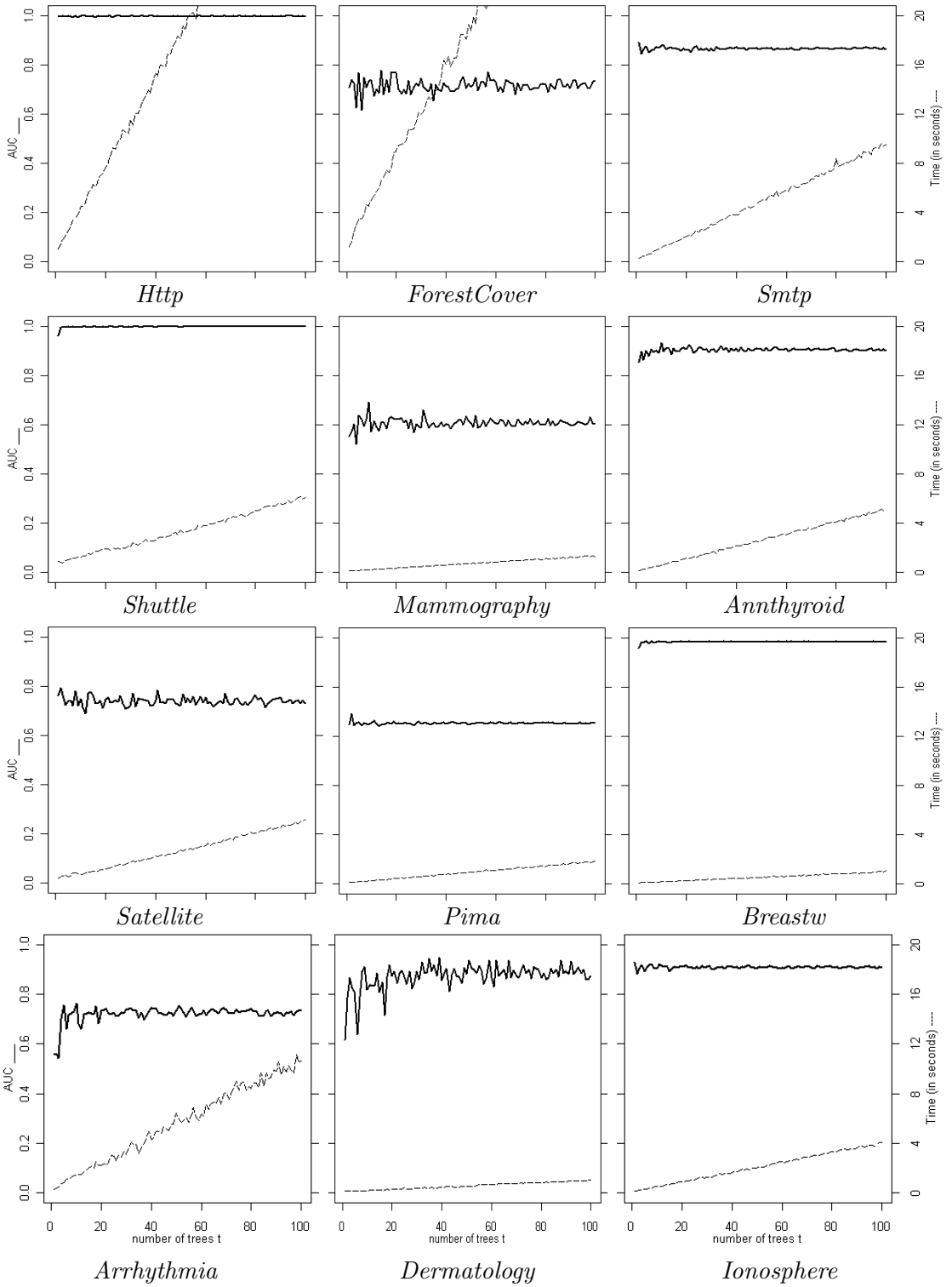


Figure C.4: SCiForest's AUC performance (y1-axis) and processing time (y2-axis) versus the different number of trees t (x-axis). Other parameters are set to $\psi = 256, q = 2$ and $\tau = 10$.

C.5 SCiForest with various sub-sampling sizes (ψ)

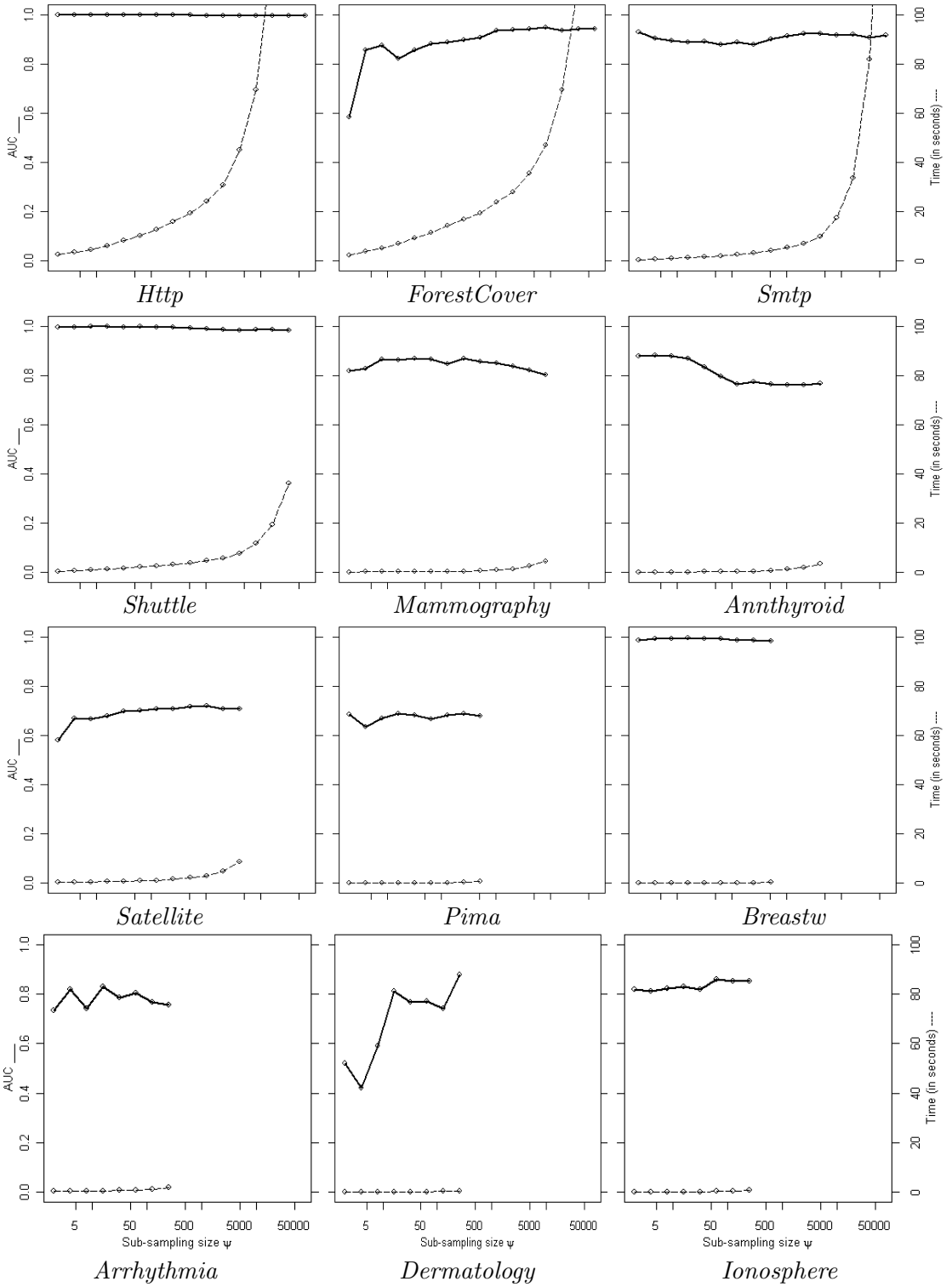


Figure C.5: SCiForest's AUC performance (y1-axis) and processing time (y2-axis) versus the different sub-sampling size ψ settings (x-axis in log scale). Other parameters are set to $t = 100, q = 2$ and $\tau = 10$.

C.6 SCiForest with various number of split selection trials (τ)

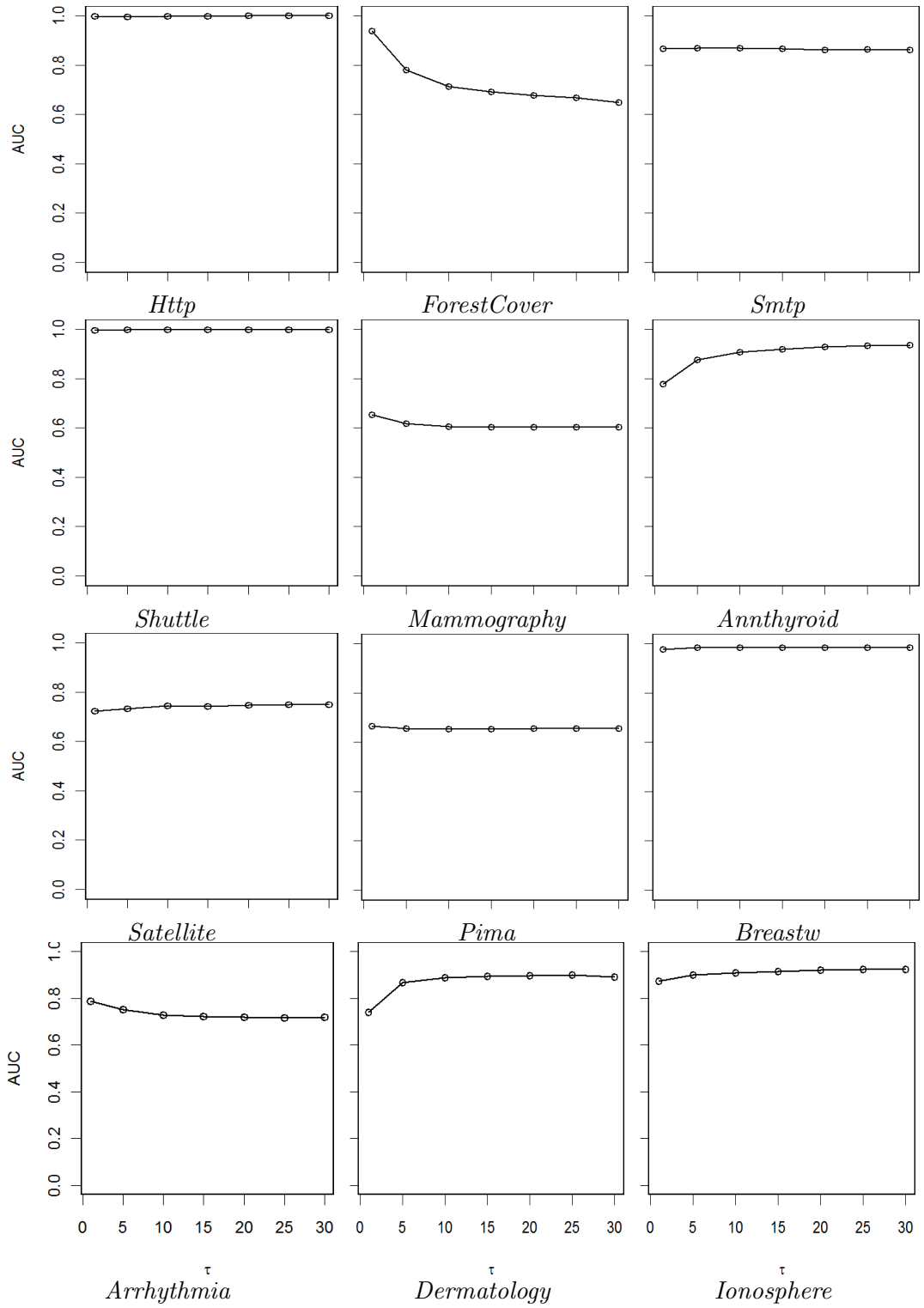


Figure C.6: SCiForest's AUC performance (y-axis) versus the different number of split selection trials τ (x-axis). Other parameters are set to $t = 100, \psi = 256$ and $q = 2$.

C.7 SCiForest with various number of attributes in hyper-planes (q)

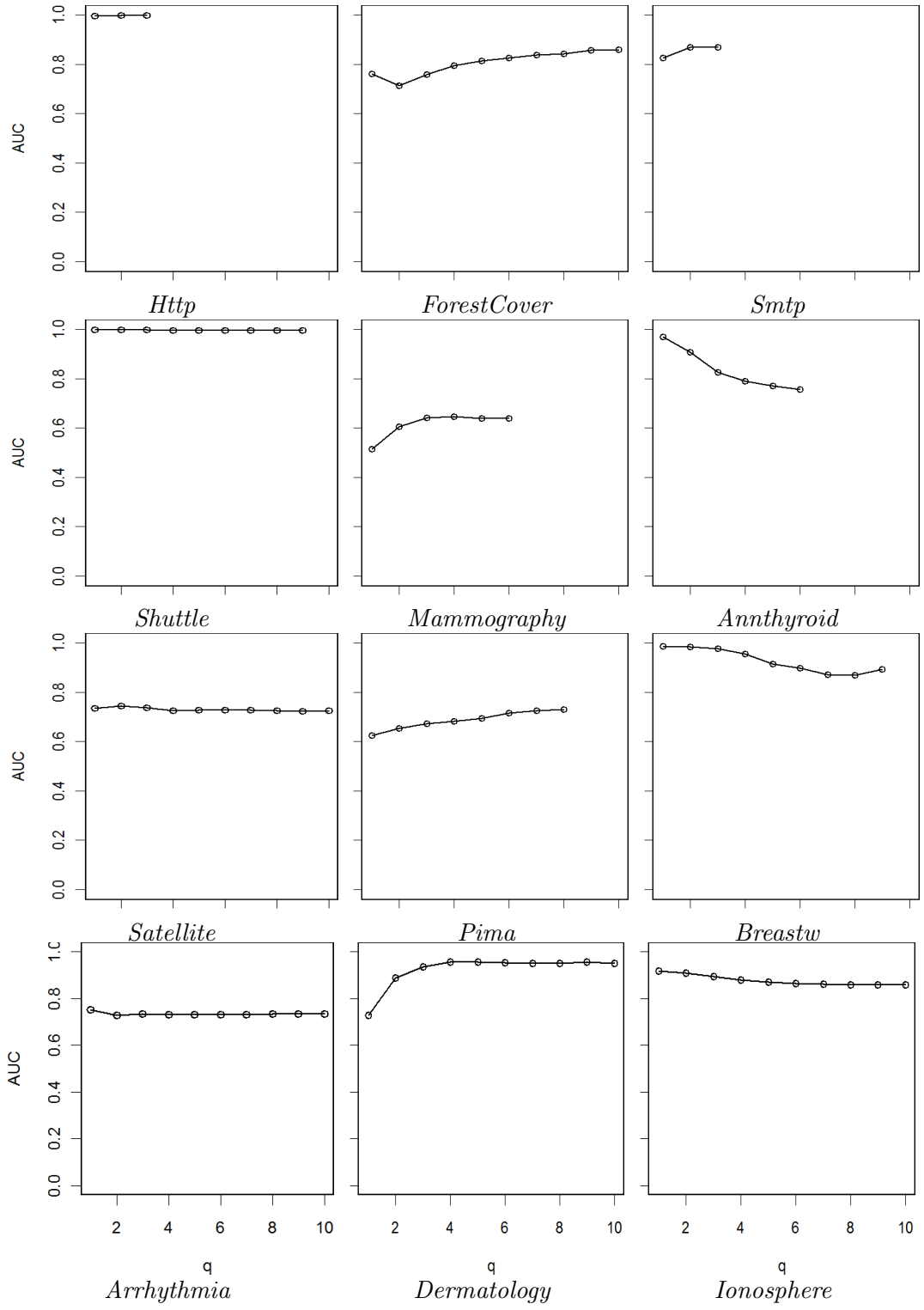
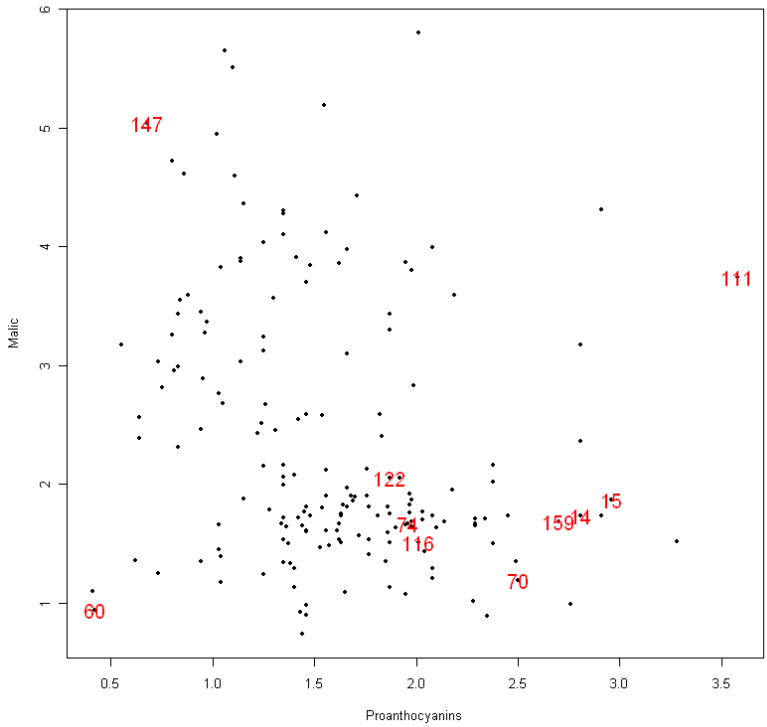


Figure C.7: SCiForest's AUC performance (y-axis) versus the different number of attributes in hyper-planes q (x-axis). Other parameters are set to $t = 100, \psi = 256$ and $\tau = 10$.

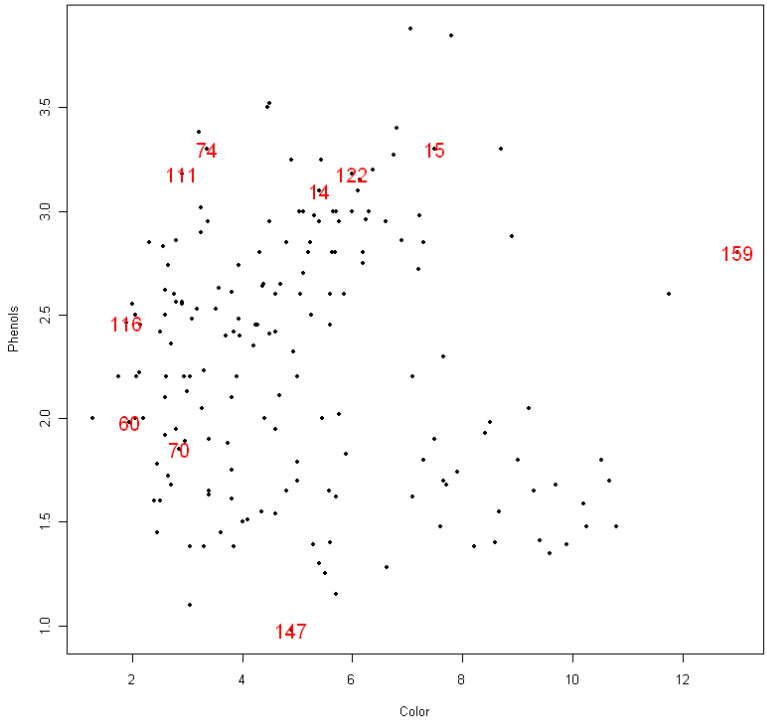
Appendix D

Visualizing Anomalies using Isolation-Based models

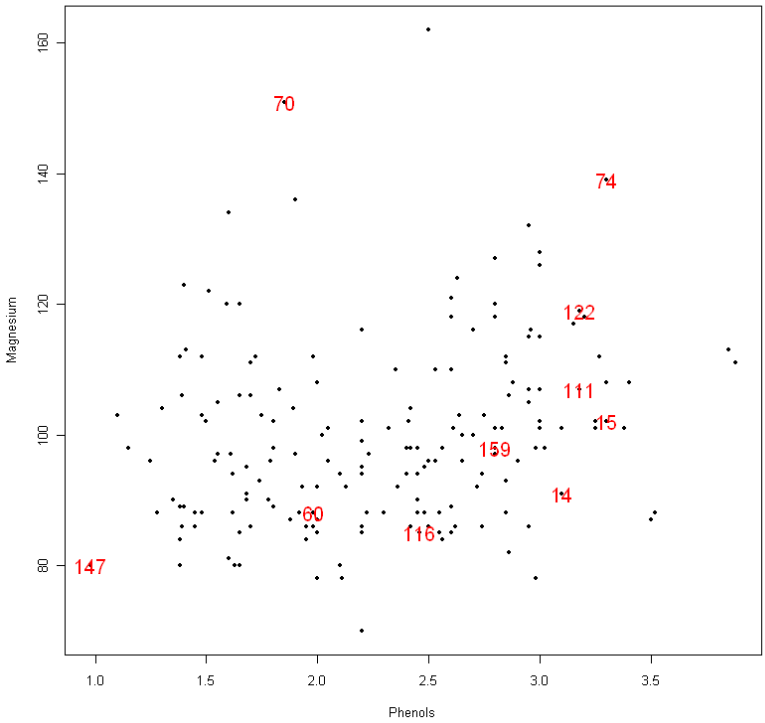
To show that the most frequent last-tested-attributes (introduced in Section 5.9) are effective in exposing anomalies using the iForest's models. Here are extra examples of using the frequent last-tested-attributes to visual anomalies. The highest-two most frequent last-tested-attributes are used as x and y axes for plotting the 2-D scatter plots. The index of the anomaly of interest is given at the bottom of each plot. Note that some of the unmarked outlying points are outlying with respect only to the attributes that are used. Marked anomalies are the top ten anomalies identified by iForest, which takes all the attributes into consideration.



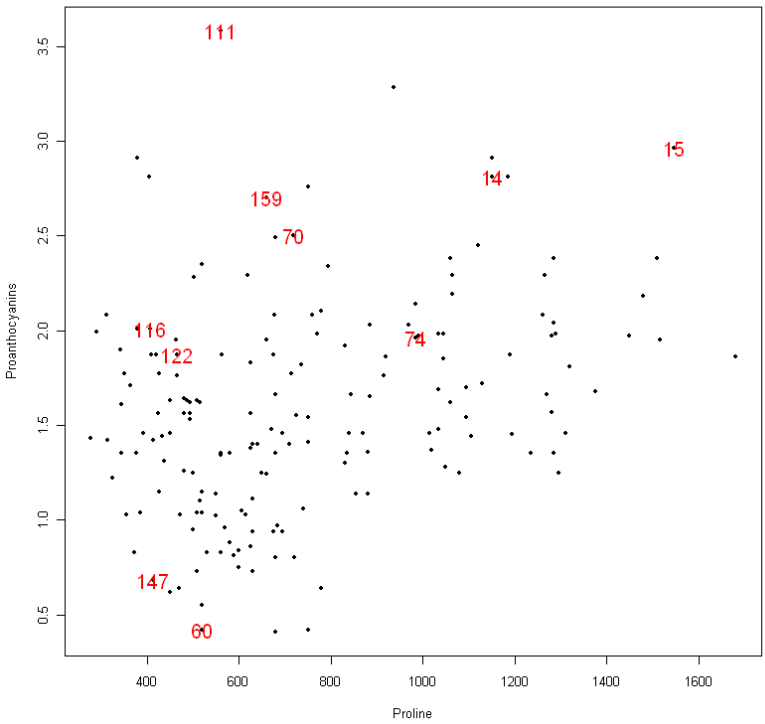
#111 (on the right)



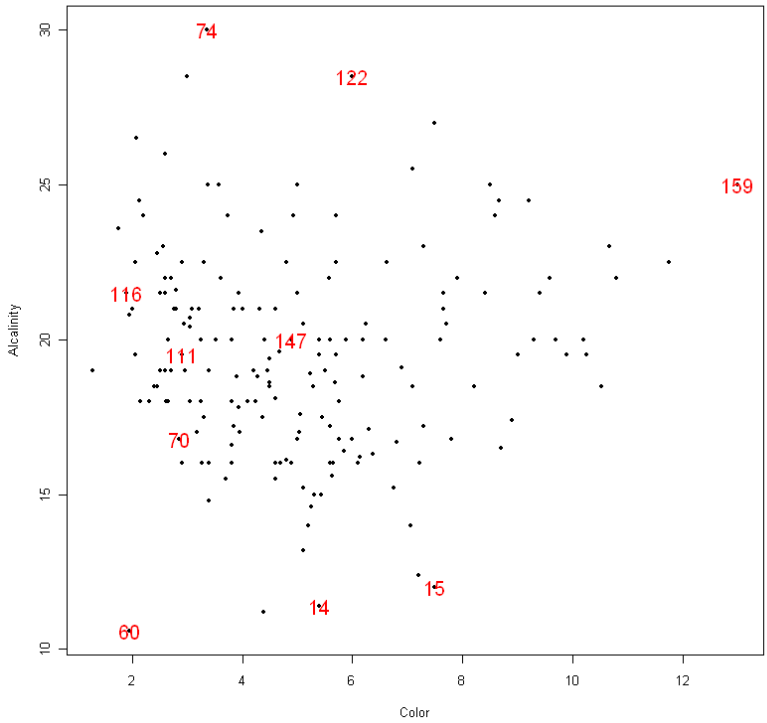
#159 (on the right)



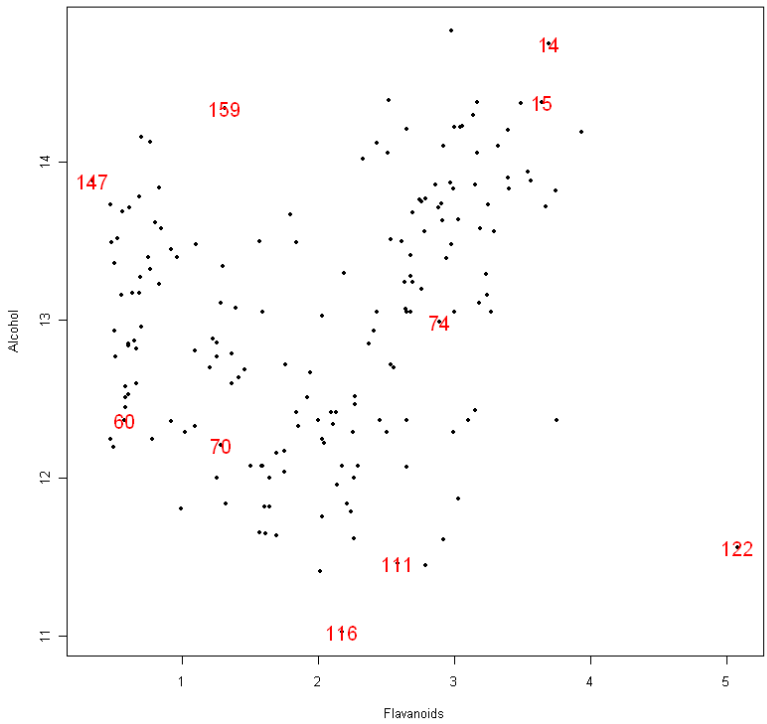
#70 (on the top)



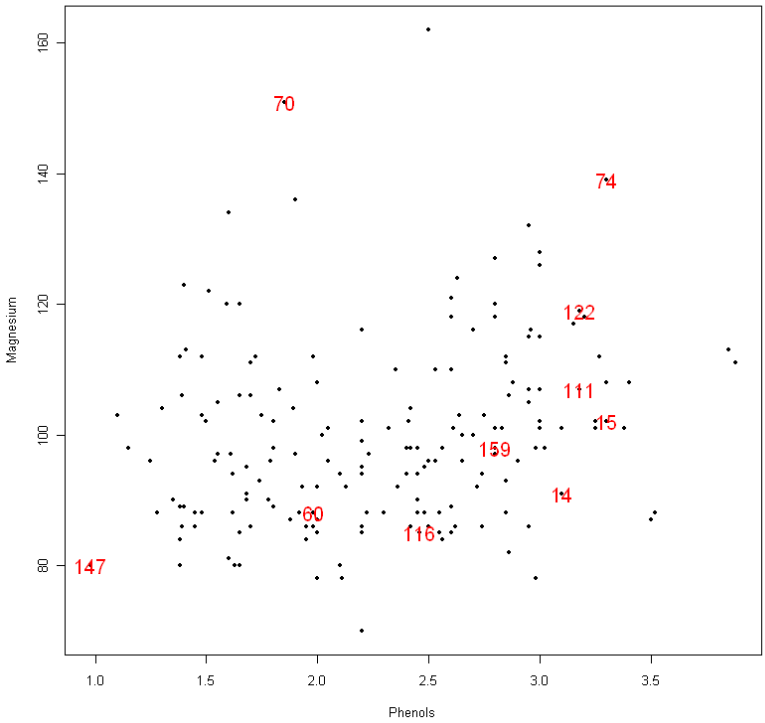
#15 (on the top right)



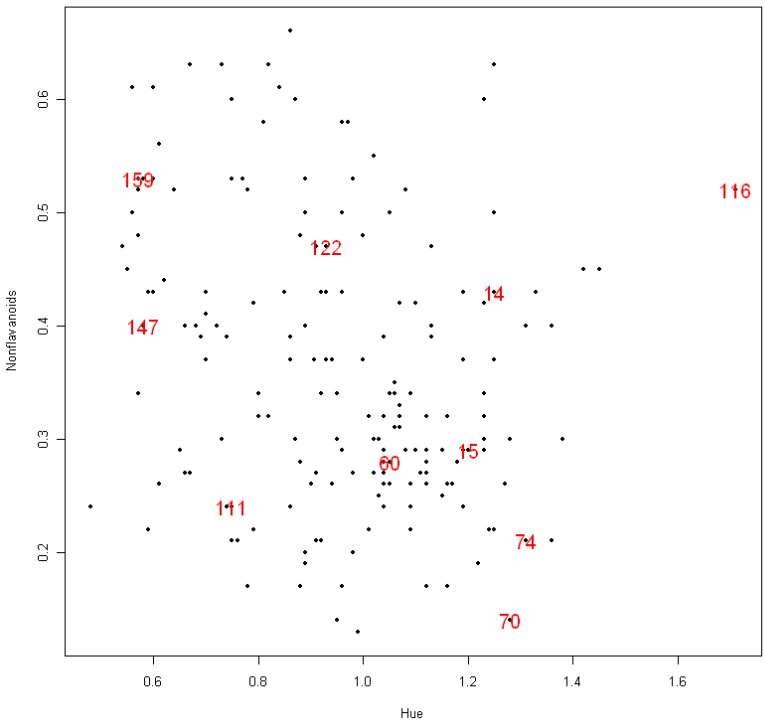
#74 (on the top left)



#14 (on the top)



#147 (on the lower left)



#116 (on the right)

Appendix E

A Preliminary Study on Categorical Data Handling

To transform categorical attributes into numeric values for isolation-based methods, we can apply the following method. In the training stage and the evaluation stage of the isolation-based methods, before we construct or evaluate an iTree, for each of the categorical attributes, we transform the values of the categorical attribute into an ordering such that the most frequent labels are placed at the centre and other less frequent labels are placed on either side. This is designed to mimic the distribution in numeric attributes, in which the most frequent points usually appear at the centre and less frequent points appear on either side of a distribution. The details of this transformation procedure can be found in Algorithm 4. In lines 1 to 3, we work out the ranking order of all the possible labels in attribute A. The for-loop in lines 5 to 8 computes the ordering of each element $\alpha \in A$. Line 6 calculates the basic ordering, which places the most frequent elements at the centre and less frequent elements on either side. The function $labelrank(\alpha, R)$ returns the label ranking of α according to R. Line 7 adds a random number $rand()$, valued between zero and one, to the basic ordering, such that elements of the same label spread within this range. This makes the final ordering compatible to numeric attributes in terms of being continuous. An illustration can be found in Figure E.1. The above transformation is done independently for the categorical attributes in each random sub-sample that constructs each iTree. The same procedure is also applied at the evaluation stage before evaluating each iTree. Although the orderings of one iTree are different from another iTree, we find

that this does not affected the path length convergence in isolation-based methods. We will provide more evidence in future studies.

Algorithm 4 : $CategoricalOrdering(A)$

Inputs: A - values of a categorical attribute, $A = \{\alpha_1, ..., \alpha_n\}$
Output: Z - ordering of A

1: let L be the possible labels of A
2: let F be the frequencies of L in A
3: let R be the ranking of L in descending order according to F
4: $Z \leftarrow \emptyset$
5: **for** $i = 1$ to $|A|$ **do**
6: $z \leftarrow \text{ceiling}(|L|/2) + ((labelrank(\alpha_i, R) \mod 2) \times 2 - 1) \times(-1)\times \text{floor}(labelrank(\alpha_i, R)/2)$
7: $z \leftarrow z + rand()$
8: $Z \leftarrow Z \cup z$
9: **end for**
10: return Z

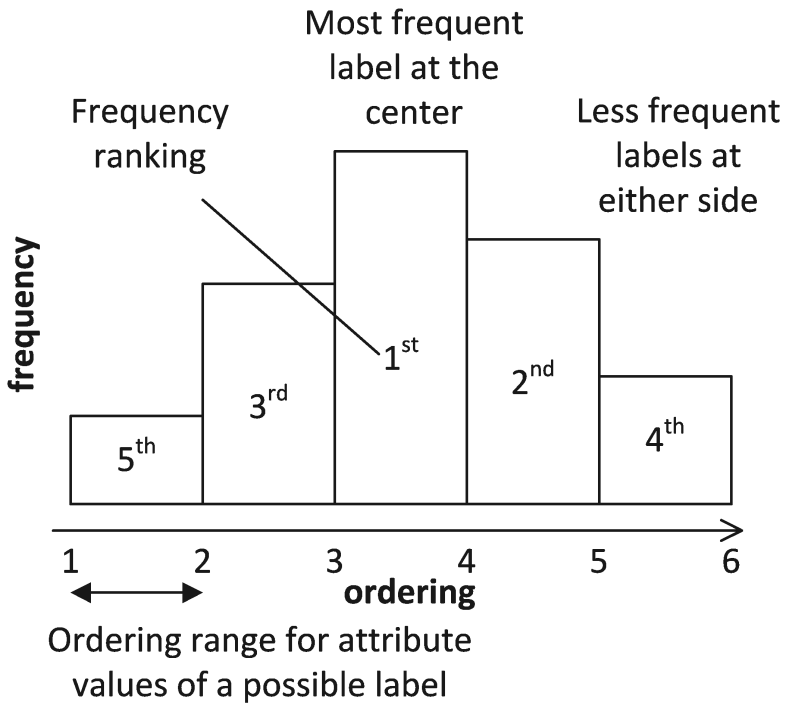


Figure E.1: A simple conversion of categorical values to numeric ordering using the frequency rankings of possible labels.

The following experiments are conducted to verify the effectiveness of this simple conversion. We do not expect the method to be the best in handling categorical data. However, we would like to show that this method is able to deal with categorical data in isolation-based methods. In Section E.1, we experiment with categorical data and in Section E.2 mixed type data.

E.1 Categorical Only Data

Our preliminary experiment uses the *Lymphography* and *BreastCancer* data sets, following their usage in (He et al., 2005a). The *Lymphography* data set has 148 instances in 18 attributes. There are 4 classes in the data and classes 1 and 4 (4.1%, 6 instances) are used as the anomalies, classes 2 and 3 (95.9%, 142 instances) are used as the normal points. The *BreastCancer* data set from (Williams et al., 2002) is a modified verion of Wincosin Breast Cancer data set (*Breastw*), in which some malignant instances are randomly removed to form an imbalanced distribution. In the *BreastCancer* data set, the class malignant (8%, 39 instances) is used as the anomalies and the class benign (92%, 444 instances) is used as the normal points. Since both data sets are small, we use a smaller $\psi = 128$ for both the SCiForest and iForest. In the following experiment, we compare SCiForest and iForest with the LSA and KNN anomaly detectors, which are introduced in the following two paragraphs.

LSA (He et al., 2005a) stands for local search algorithm; it minimizes the entropy of the presumed normal set using multiple iterations. In each iteration, points in the presumed normal set are swapped with points in the anomaly set to minimize the objective. The complexity of LSA is $O(nbdI)$, where b is the number of anomalies and I is the number of iterations. LSA is only suitable for small data sets.

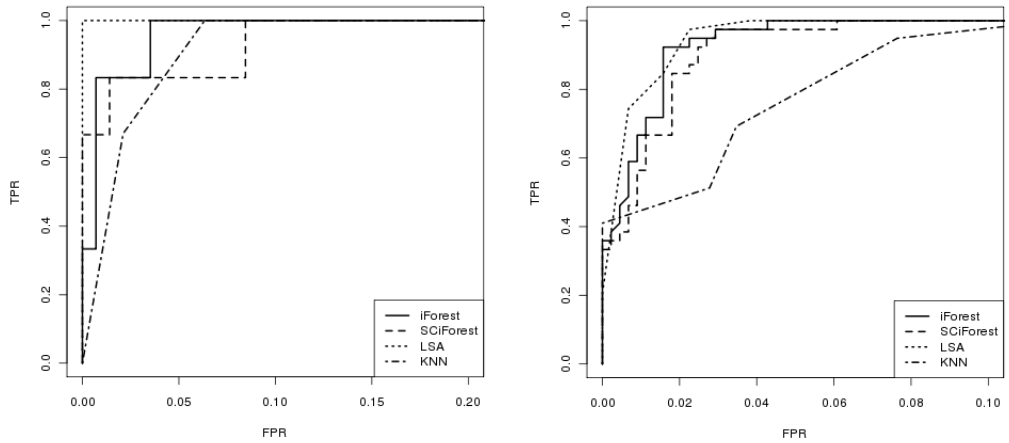
KNN (Ramaswamy et al., 2000) uses the distance of the k^{th} nearest neighbour as the anomaly score. In categorical data, the distance between two instances can be measured by:

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{sim(\mathbf{x}_i, \mathbf{x}_j)} - 1$$

The similarity between two instances is usually a summation of individual attribute similarities between the two instances $sim(\mathbf{x}_i, \mathbf{x}_j) = \sum_{g=1}^{d} sim_g(\mathbf{x}_i, \mathbf{x}_j)$. A simple way to measure the similarity $sim_g(\mathbf{x}_i, \mathbf{x}_j)$ between two instances $\mathbf{x}_i$, $\mathbf{x}_j$ is to assign the value 1 if the g^{th} attribute values of the two instances are the same and 0 otherwise (Boriah et al., 2008).

Using the detection result in (He et al., 2005a), we are able to reconstruct the ROC curves for comparison. Although the ROC curves may not be as detailed, they are sufficient for the purpose of this experiment. Figure E.2 illustrates the ROC curves of SCiForest, iForest, LSA and KNN and we find that SCiForest and iForest are comparable to LSA

and KNN. The differences between different these detectors are very small and we only show a narrow range of false positive rate in Figure E.2.



Lymphography *BreastCancer*

Figure E.2: The ROC curves of four detectors on two data sets. In detecting anomalies, iForest and SCiForest have a similar performance as compared to LSA and KNN, for categorical data,. Note that only a narrow range of false positive rate (FPR) is shown.

E.2 Mixed Type Data

In the following experiment, we are going to experiment with the categorical data conversion on mixed type data. Because these data sets do not have the ground truth anomalies, we opt to showcase interesting examples within the top 10 anomalies detected by iForest.

- **Adult**
  - `age:59, Gender:male, race:pacific-islander, native-country:china, education:5th-6th grade, occupation:exec-managerial.`

    (This record is unique in the combination of race, native-country, education and occupation. It shows that iForest is effective in selecting a rare instance that is very different from the others.)

- **Abalone**
  - `sex:I, length:0.075, diameter:0.055, height:0.010, whole:0.002, shucked:0.001, viscera:0.001, shell:0.0015, rings:1.`

(This is the smallest, lightest abalone with the least number of rings in the database.)

– `sex:F, length:0.815, diameter:0.650, height:0.250, whole:2.255, shucked:0.8905, viscera:0.42, shell:0.7975, rings:14.`
(This is the longest, widest and heaviest abalone in the database.)

The above results show that iForest is able to retrieve some interesting records in mixed type environments.

Vita

Publications arising from this thesis include:

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, Isolation-based Anomaly Detection, Transactions on Knowledge Discovery from Data (TKDD) (Accepted with minor revisions on the 23^{rd} July 2010).

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, 2010, *On Detecting Clustered Anomalies using SCiForest*, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010).

Kai Ming Ting, Guang-Tong Zhou, Fei Tony Liu and James Tan Swee Chuan, 2010, *Mass Estimation and Its Applications*, ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010).

Guang-Tong Zhou, Kai Ming Ting, Fei Tony Liu and Yilong Yin, 2010, *Relevance Feature Mapping for Content-Based Image Retrieval*, Multimedia Data Mining Workshop at KDD 2010.

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, 2008, Isolation Forest, IEEE International Conference on Data Mining (ICDM 2008).

Award arising from this thesis:

Best Paper Award (Runner up) in the IEEE International Conference on Data Mining 2008 (ICDM 2008).

Permanent Address: Gippsland School of Information Technology

Monash University

Australia

This thesis was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Glenn Maughan and modified by Dean Thompson and David Squire of Monash University.