Why is this an Anomaly? Explaining Anomalies using Sequential Explanations

UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

A dissertation submitted in partial fulfilment of the requirements for the degree Master of Science.

Author: Tshepiso Mokoena (371416)

Supervisor: Prof. Turgay Celik

Co-Supervisor: Dr. Vukosi Marivate

School of Computer Science and Applied Mathematics,
Faculty of Science,
University of the Witwatersrand, Johannesburg.

October 2019

Abstract

Anomaly detection has received much attention throughout the years. Currently, human analysts in real-world applications use anomaly detectors to assist them in identifying potential anomalous data points. Unfortunately, most anomaly detectors do not provide the analysts with explanations about what makes a data point anomalous, resulting in the analysts to consider the information related to the entire feature space of each detected data point to decide whether they are truly anomalous or not. This process can be time-consuming and costly in most domains, especially if the feature space is large, and feature interactions are critical to the analyst's judgement. To assist the analyst and minimise the number of features that they must analyse to identify true anomalies confidently, we introduce an explanation called a Sequential Explanation(SE). A SE for a detected data point contains subsets of features that explain why the detected data point could be anomalous to the analyst. The first subset in the SE contains only one feature; the second subset contains two features; the third subset contains three features, and so on. The subsets of features explain to the analyst why the detected data point could be anomalous. The subsets of features in the SE are incrementally presented to the analyst one at a time, in order, until the analyst has acquired enough information to decide whether the data point is an anomaly or not. In this thesis, we introduce two novel methods of generating SEs that will work alongside any anomaly detector. The first method is the outlier-based method that adds features in the SE by using an anomaly detector's outlier scoring measure guided by a search algorithm. The sample-based method uses sampling to turn the problem into a classical feature selection problem such that any feature selection algorithm can be used to generate the SE. In our experiments we (i) analyse the performance and complexity of different anomaly detectors' outlier scoring measures and search algorithms in the outlier-based SEs, (ii) analyse the performance and complexity of different feature selection methods in the sample-based SEs and (iii) compare the outlier and sample-based SEs based on their performance and complexity. In addition we also introduced a new and improved method of evaluating explanations called the area under the curve of the analyst certainty curve (AUCC). Our results show that both the outlier and sample-based methods can generate SEs which significantly outperform randomly presenting features to the analyst. In conclusion, we found that our SEs were able to identify the features that explain the anomalies and that our new evaluation method is an improvement on the previous evaluation method used to evaluate explanations.

Declaration

I, Tshepiso Mokoena (student number: 371416), declare that this research project is my own work, and all information sources have been referenced. Additionally, this work has never been submitted to any other university for any other degree.

Signed: _____

Date: _____

Acknowledgements

First and foremost, I would like to thank God for seeing me through this period of my life.

To my two supervisors, Prof. Turgay Celik and Dr. Vukosi Marivate, thank you for your constant support and belief in me even when I saw no light at the end of the tunnel.

I would also like to thank my family and friends for their constant love and encouragement throughout.

Publications

Part of this work has been presented at the following conferences:

1. Mokoena, T., Lebogo, O., Dlaba, A. & Marivate, V. (2017), Bringing sequential feature explanations to life, in '2017 IEEE AFRICON', IEEE, pp. 59–64.

2. Mokoena, T., Marivate, V. & Celik, T. (2016), Tackling imbalanced data sets for sequential feature explanation generation using cost sensitive learning and sampling, in 'Australasian Data Mining Conference (AusDM 16)', pp 166 – 173. **URL**: ausdm.org/ausdm16.pdf.

We are also in the process of submitting the following journal:

1. Mokoena, T., Marivate, V. & Celik, T. (2019), Why is this an Anomaly? Explaining Anomalies using Sequential Explanations , Elsevier.

Contents

List of Figures

List of Tables

Abbreviations

AD: Anomaly detector

AUCC: Area under the analyst certainty curve

BS: Backward selection

BS-SVM: Backward selection support vector machine

EGMM: Ensemble Gaussian Mixture Model

FS: Forward selection

FS-SVM: Forward selection support vector machine

GMM: Gaussian Mixture Model

IND-BS: Independent backward selection

IND-FS: Independent forward selection

iTree: Isolation forests

KDE: Kernel density estimation

LOF: Local outlier factor

MFP: Minimum feature prefix

MIFS: Mutual information feature selection

mRmR: Maximum relevance minimum redundancy

OCSVM: One-class support vector machine

PSO: Particle swarm optimisation

RF: Random forest

RF-RFE: Random forest recursive feature elimination

SFE: Sequential feature explanation

SE: Sequential explanation

SVM: Support vector machine

SVM-RFE: Support vector machine recursive feature elimination

Chapter 1

Introduction

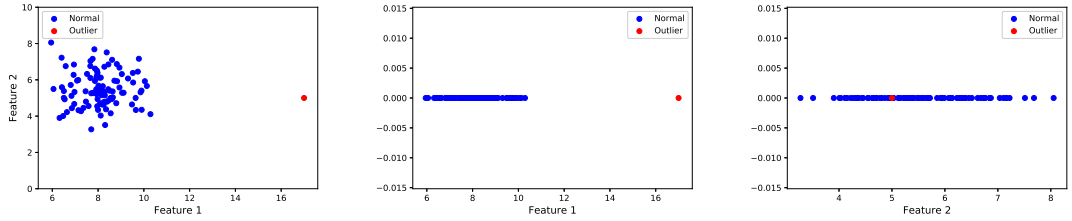
In this chapter, we introduce our research work, explain why it is important with regards to the problem we are solving. Section 1.1 provides a background of our research and why it is important, Section 1.2 defines our problem statement, Section 1.3 describes our research objectives, Section 1.4 defines the research questions we want to answer in our work, Section 1.5 details the contributions that will come from our study and lastly, Section 1.6 provides an overview of the rest of the thesis.

1.1 Background

Anomalies are known as rare data points that are generated by a process that is distinct from the process generating the "normal" data points (Siddiqui et al. 2015, 2019). Anomaly detection is the problem of identifying the process that generates the anomalies within a dataset. The field of anomaly detection spans a vast number of research areas such as Machine learning, Data mining, Statistics, Information theory, Spectral Theory and many more (Chandola et al. 2009). A variety of application domains have applied anomaly detection over the years. These application domains include but are not limited to the following (Chandola et al. 2009): intrusion detection (Lee et al. 1998, Portnoy et al. 2001, Garcia-Teodoro et al. 2009, Liao et al. 2013), fraud detection (Phua et al. 2010, Ngai et al. 2011, Kirkos et al. 2007), medical and public health (Wong et al. 2002, Polat et al. 2005, Greensmith et al. 2006), industrial damage detection (Alzghoul & Löfstrand 2011, Zhang et al. 2008, Xue et al. 2006, Basu & Meckesheimer 2007), image processing (Matteoli et al. 2010, Tarabalka et al. 2009), text data (Manevitz & Yousef 2001, Anantharam et al. 2012, Miller et al. 2014, Laorden et al. 2014) and sensor networks (Rajasegarar et al. 2006, Du et al. 2006, Zhang et al. 2010). An anomaly in any one of these domains could represent an interesting phenomenon or a severe cause of

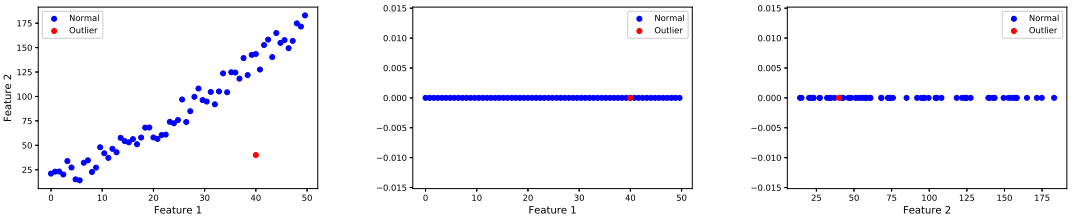
concern which will need to be addressed urgently by a human analyst analysing the data. For example, an anomaly in insurance claims data could represent fraudulent claims, whereas an anomaly in astronomical data could represent discoveries of new galaxies or interesting new planets. A variety of anomaly detection methods are used by analysts in these domains to identify anomalies. In most cases the process generating the anomalies is unknown, resulting in anomaly detection methods in real-world applications to operate in an unsupervised mode by looking for outliers in the dataset with the hope that they are anomalies (Siddiqui et al. 2015, 2019, Emmott et al. 2015). An outlier is a data point which deviates so much from the rest of the other data points in the dataset that it arouses suspicions that a different process generated it. However, outliers not always equate to anomalies. For example, a computer security application of a company may exhibit an unusually high amount of copying and printing activities at the end of the month. These activities could cause a reason for concern because of the high amounts of copying and printing which are not normal compared to other days of the month, therefore, making them appear highly outlying and classified as anomalous activities. However, because these activities occurred at the end of the month and there are month end processes such as payrolls and monthly audits that need to be copied and printed, then there is a valid explanation behind the high amounts of copying and printing that exempt them from being anomalous. Due to this, a human analyst is required to investigate the outliers to decide which ones are most likely to be true anomalies and deserve further action.

An analyst provided with an outlier point is faced with the challenge of identifying the features associated with the outlier that makes the outlier outlying to decide whether it is an anomaly or not. In most cases, the analyst is forced to manually browse through the full feature space of the outlier point to identify the outlying subset of features that will give them enough information to classify whether the outlier point is truly anomalous or not. The identification of the outlying subset of features can be extremely challenging even if tens of features characterise the feature space, especially when feature interactions are critical in the decision making. For example, Figures 1.1 (a) and 1.2 (a) show different scenarios of outliers in two-dimensional feature space datasets. For each figure, the red data point represents an outlier detected using an arbitrary anomaly detector, while the blue data points represent the normal data points. We observe from Figure 1.1 (a) that the outlier point is outlying because it does not form part of the cluster of blue data points, whereas from Figure 1.2 (a) we observe that the outlier is outlying because it does not form part of the linear trend that is followed by the blue data points. An analyst analysing these outlier points would want to know along which features the outliers are outlying to make a judgement on whether they are truly anomalous or not. Figures 1.1 (b) and (c) show visualisations of the outlier in



(a) (b) (c)

FIGURE 1.1: Plot (a) shows an outlier detected by an arbitrary anomaly detector in a 2-dimensional feature space, while plots (b) and (c) show the detected outlier visualised in Features 1 and 2 respectively. It is clear from plots (b) and (c) that the outlier detected in the plot (a) is outlying only in Feature 1.



(a) (b) (c)

FIGURE 1.2: Similar to Figure 1.1, however, the detected outlier in this case is not outlying in Feature 1 or Feature 2. The outlier is outlying due to the feature interactions of Feature 1 and Feature 2

Figure 1.1 (a) in Features 1 and 2 respectively. We observe from Figure 1.1 (c) that the outlier appears normal in Feature 2, whereas, from Figure 1.1 (b) the outlier is outlying in Feature 1. Therefore, Feature 1 is the outlying feature subspace that the analyst must identify to decide whether it provides enough information to classify the outlier as anomalous or not. On the other hand, we observe from Figures 1.2 (b) and (c) that the outlier in Figure 1.2 (a) is not outlying when considering Features 1 and 2 independently. The outlier is only outlying due to the feature interactions of Feature 1 and Feature 2 in Figure 1.2 (a).

Figure 1.3 presents the anomaly detection pipeline in real-world applications (Siddiqui et al. 2019). First, an analyst uses an anomaly detector to identify outliers in a dataset that contains normal and anomalous data points. From the detected outliers, the analyst has to analyse further which ones are truly anomalies, whereas the analysts discard the data points which are considered to be non-outliers by the anomaly detector. There are two significant risks which we observe from this pipeline. The first risk is from the anomaly detector, while the second risk is from the analyst. The first risk involves the outliers detected by the anomaly detector which may contain anomalies and false positives (missed true normals) while the data points that the anomaly detector considers to be normal may contain normal data points and false negatives (missed true anomalies).

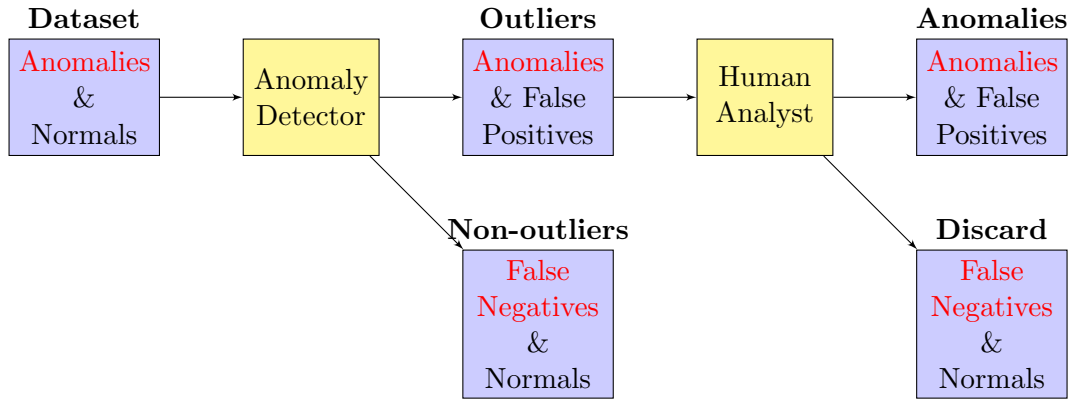


FIGURE 1.3: Anomaly Detection Pipeline (Siddiqui et al. 2019)

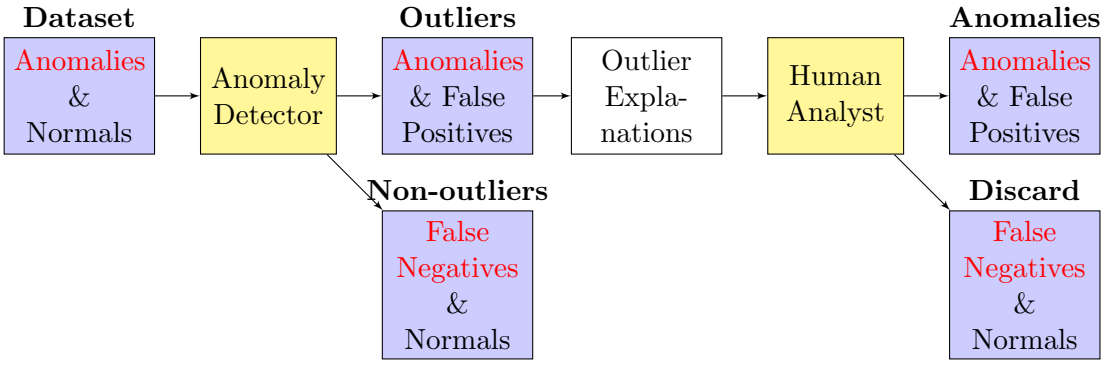


FIGURE 1.4: Anomaly Detection Pipeline with outlier explanations (Siddiqui et al. 2019)

In this case, the anomaly detector runs the risk of discarding true anomalies, which it considers to be normal. The second risk involves the outliers presented to the analyst. In this case, the analyst runs the risk of misidentifying true anomalies and discarding them as normal and misidentifying normal data points and classifying them as anomalous. There has been a vast amount of work done in the anomaly detection domain, therefore, mitigating the first risk can be done by reducing anomaly detectors false positive and false negative rates by either re-calibrating the anomaly detection model or changing it. Making it easier for the analyst to identify true anomalies can mitigate the second risk. In this thesis, the focus is to compute outlier explanations that explain to an analyst what makes an outlier point outlying. An outlier explanation is a subset of features from the feature space of the outlier that will give an analyst enough information to classify whether an outlier point is truly anomalous or not by using their domain expertise. Our work focuses on improving the anomaly detection pipeline for the analyst by providing them with outlier explanations. Figure 1.4 shows the new anomaly detection pipeline, which includes outlier explanations for the analyst.

Siddiqui et al. (2015, 2019) developed an outlier explanation called a sequential feature

explanation (SFE). An SFE for a detected outlier point is an ordered sequence of features, where the order of the features in the SFE indicates their importance to explaining to the analyst what makes the outlier outlying. An SFE for a detected outlier point incrementally presents the outlying features of the outlier point to the analyst by first presenting the first feature in the SFE. If the analyst can decide that the outlier point is anomalous or not with only that one feature presented, then the analyst's job is done. If the analyst cannot make a decision based on the single feature presented, then the next feature in the SFE is presented. Now the analyst has seen the two most important features that explain why the outlier point is outlying. The process of incrementally adding features to the set of features already presented to the analyst will continue until the analyst has acquired enough information to decide whether the outlier point is an anomaly or not. The SFE aims to (*i*) identify the subset of outlying features that were responsible for the detection by the anomaly detector and (*ii*) minimise the number of features that have to be revealed for the analyst to identify true anomalies confidently.

Evaluations of SFEs are done by simulating an analyst and comparing the minimum feature prefix (MFP) (Siddiqui et al. 2015, 2019). The idea of the approach is to construct a simulated analyst for each anomaly detection benchmark using supervised learning on the ground truth about which points are anomalies and normal. The simulated analyst provides a probability score of how anomalous a data point is given the features presented to it. The MFP is the minimum number of features required for the simulated analyst to reach a probability threshold of how anomalous a data point is given the number features presented in the SFE. For example, if the probability threshold is 0.8, then the MFP is 4 if the probability of the first feature presented to the analyst from the SFE is 0.1, 0.3 for the next two features presented, 0.7 for the next three features presented, 0.81 for the next four features presented. The MFP is 4 because the probability threshold reached 0.80 after the presentation of the fourth feature to the simulated analyst. A data point is classified as an anomaly when it reaches the MFP threshold. The lower the MFP of an SFE, the better it is.

1.2 Problem Statement

Outlier explanations aim to make it easier and minimise the effort required for the analyst to identify true anomalies during the investigation. Without any assistance, the analyst will be required to manually browse through the feature space of each detected outlier point to identify the outlying features that they can use to make their judgements about which outlier points are truly anomalous. In most domains, manually browsing

through the feature space is practically infeasible and increases the chances of the analyst not identifying true anomalies, which can be extremely costly.

The investigative work of the analyst is related to the number of features that must be revealed before they can make their decision. The sooner they can make their decision on an outlier point, the sooner they can take action and move on to analysing more outlier points. SFEs are the first form of outlier explanations to consider incrementally revealing the outlying features to the analyst until they can make a decision. The sooner the analyst can make a decision, the better. Thus the goal of the SFE is to present the features of each outlier incrementally where the order of the presentation of the features indicates their importance to explaining to the analyst what makes each outlier outlying.

The current SFE computation methods developed by Siddiqui et al. (2015, 2019) only work for density-based anomaly detectors that meet the assumption that they can compute outlier scores for an outlier point in any given subset of features. Siddiqui et al. (2015, 2019) used an Ensemble Gaussian Mixture Model (EGMM) (Glodek et al. 2013) as their anomaly detector and used the marginal densities of the model which are straightforward to derive in closed forms (Siddiqui et al. 2015, 2019) to compute the outlier scores for an outlier point in any given subset of features. With this assumption, the SFEs compute explanations that explain why the EGMM judged an outlier point to be outlying and not why the outlier is outlying in general. In practice, most anomaly detectors do not provide these scores trivially. It then makes it impossible to compute SFEs that explain why the particular anomaly detector judged an outlier point to be outlying except for density-based anomaly detectors such as the EGMM used by Siddiqui et al. (2015, 2019). The dependence of the assumption on the anomaly detector is one of the disadvantages of SFEs because computing the outlier scores in any given feature space in most anomaly detectors is a non-trivial task. Another disadvantage of SFEs is their sequential greedy nature which results in an effect known as the "nesting effect". Once a feature is added into the SFE, it cannot be removed, and this could provide sub-optimal explanations to that analyst. For example, the subset of the five best features in the SFE must contain the subset of the one, two, three and four best-chosen features in the SFE. In practice, the five best features may not contain any of the best one, two, three or four best-chosen features (Nakariyakul & Casasent 2009) and this could lead to the analyst making a wrong decision or requiring the presentation of more features from the SFE to make their final decision.

MFPs are used to evaluate SFE methods. However, the first problem we observe regarding the evaluation of SFEs using MFPs is that two or more explanations with the same MFP could be considered to be equal even if one of the SFEs returns significantly higher probabilities than the other. For example, if the first four features in one

SFE returns probabilities of $0.1, 0.3, 0.5, 0.82$ and another SFE returns probabilities of $0.7, 0.75, 0.78, 0.95$, then they both have an MFP of 4 if the analyst's threshold is 0.8. However, the second SFE is far more superior than the first SFE because, for every sequential feature presented to the simulated analyst, it always returns higher probabilities. The second problem is that the supervised learning model used to generate the probabilities of the simulated analyst does not take class imbalances into account. Anomaly detection benchmarks are imbalanced because the anomalies are generally significantly fewer than the normal data points, therefore fitting a model that does not take class imbalance into account introduces a bias towards the normal class of data points (Krawczyk 2016). The third problem we observed is that the outlier point whose probability is being predicted in the given feature space is used to train the simulated analyst model. Including the data point that requires a prediction in the training set of the model introduces a bias towards its probability prediction.

1.3 Research Objectives

In our problem statement, we identified the current problems associated with SFEs and in their evaluations. In this work, we have two research objectives. The first objective is to improve the current state of outlier explanations by introducing a new explanation called a sequential explanation (SE) that address the shortcomings of the SFE, while the second objective is to introduce a new evaluation method called the area under the analyst certainty curve (AUCC) that addresses the shortcomings of the MFP.

1.3.1 Sequential Explanations

Similar to an SFE, an SE for a detected outlier point is presented sequentially to the analyst. The SE sequentially presents the best feature subsets in increasing size. The SE first presents the analyst with the first feature subset containing one feature. If the analyst can decide that the outlier point is anomalous or not based on that one feature, then the analyst's job is done. If the analyst cannot make a decision based on the single feature presented, then the next feature subset containing two features in the SE is presented. The process of presenting the analyst with the best feature subsets in increasing size will continue until the analyst has acquired enough information to decide whether an outlier point is an anomaly or not.

The SE has several advantages over the SFE. Firstly, an SE can be used as an additional step for any anomaly detection algorithm because it does not have knowledge of the anomaly detection algorithm and treats it as a black-block and explains which features

make its detected outliers outlying. Secondly, it does not suffer from the nesting effect. For example, the third feature subsets in the SE, which contains three features, does not have to be a subset of the subsequent feature subsets in the SE. The third advantage SEs have over SFEs is that they do not enforce the SFE assumption which restricted SFEs to only work on density based anomaly detectors that can compute outlier scores for an outlier point in any given subset of features.

In our work, we develop two novel methods of computing SEs called the sample and outlier-based SEs. Both the sample and outlier-based SEs make use of greedy nested sequential search methods and a non-nested meta-heuristic search method to compute the SEs. Sample-based SEs turn the problem of computing SEs into a classical supervised feature selection problem by using a sampling method to create a balanced inlier and outlier class for the outlier point whose SEs we want to compute. The sample-based SEs for the outlier point are the features that best separate its inlier and outlier class. Outlier-based SEs compute SEs by using an anomaly detector's outlier scoring measure guided by a search algorithm to identify the features that make the outlier point of interest outlying based on the outlier scoring measure used.

The sub-objectives we want to achieve through the SEs are as follows:

1. Determine if the sample and outlier-based SEs provide outlier explanations that are better than the analyst manually browsing for the outlying feature subsets.
2. Compare the performances of the different anomaly detector's outlier scoring measures and search methods for the outlier-based SEs.
3. Compare the performances of different feature selection methods for the sample-based SEs.
4. Compare the performances of the greedy nested sequential search methods and the non-nested meta-heuristic search method.
5. Compare the performance of the sample and outlier-based SEs.
6. Do a complexity analysis of the outlier and sample-based SEs.

1.3.2 Area under the analyst certainty curve

To address the first problem of the MFPs where two or more explanations with the same MFP are considered to be equal even if one of the explanations returns significantly higher probabilities than the other, we will use a new measure called AUCC and show that it performs better than the MFP. To address the second and third problems, we will

use a supervised learning model that takes class imbalances into account to generate the simulated analyst and remove the outlier point whose probabilities we want to predict from the training set to obtain unbiased probabilities from the simulated analyst.

The sub-objectives we want to achieve through the AUCC are as follows:

1. Compare the evaluation measure of the MFP and AUCC.
2. Determine the advantages that the AUCC and the analyst certainty curves have over the MFP.

1.4 Research Questions

1. For outlier-based SEs, there exists a variety of different anomaly detection algorithms that score outliers by using specific properties and assumptions. Also, there is a variety of different search methods that can be used to prune the search space of the SE and avoid an exhaustive search. The different pairing combinations of anomaly outlier scoring measures and search methods lead us to the following research questions:
   (a) Do the outlier-based SEs provide explanations that are better than the analyst manually browsing for the outlying feature subsets?
   (b) Which anomaly detector's outlier scoring measure computes the best outlier-based SEs?
   (c) Between the greedy nested sequential search methods and the non-nested meta-heuristic search method, which one computes the best outlier-based SEs?
   (d) Which anomaly detector's outlier scoring measure paired with a search method computes the best outlier-based SE?
   (e) What is the complexity of the different anomaly detector's outlier scoring measures and search methods?
2. For sample-based SEs, there are a variety of different feature selection methods we can use to compute the SEs. Each feature selection method uses specific properties and assumptions about the data to perform feature selection. Given a detected outlier data from an arbitrary anomaly detector, we want to use different feature selection methods to identify which ones provide us with the best sampled based SEs. We are interested in the following research questions:

   (a) Do the sample-based SEs provide explanations that are better than the analyst manually browsing for the outlying feature subsets?
   (b) Which feature selection method computes the best sample-based SEs?
   (c) Between the greedy nested sequential search methods and the non-nested meta-heuristic search method, which one computes the best sample-based SEs?
   (d) What is the complexity of the different feature selection methods?

3. Between the sample and outlier-based SEs, which method computes the best SEs?
4. Is the AUCC evaluation measure better than the MFP evaluation measure?

1.5 Research Contribution

In this thesis, we make the following contributions to the field of outlier explanations:

- Introduce a new outlier explanation called a SE that addresses the shortcomings of the SFE and treats anomaly detection algorithms as a black-block and explains which features make detected outliers outlying.
- Introduce two different methods of computing SEs called the sample and outlier-based SEs and show that they provide outlier explanations that are better than the analyst manually browsing for the outlying feature subsets.
- Compare and contrast the different anomaly detector's outlier scoring measures and search methods for the outlier-based SEs.
- Compare and contrast the different feature selection methods for the sample-based SEs.
- Provide complexity analyses of the sample and outlier-based SEs.
- Introduce a new evaluation method that addresses the shortcomings of the MFP.

1.6 Thesis Overview

The organisation of the thesis as follows, Chapter 2 gives a background of anomaly detection algorithms and a literature review of the field of outlier explanations and similar works. Chapter 3 presents the research methodology that we use in our experiments. Chapter 4 presents the analysis and discussions of the experimental results. Chapter 5 contains the conclusion of our work.

Chapter 2

Background and Literature Review

The main aim of our study is to develop SEs that will explain the outputs of any arbitrary anomaly detector. In this chapter, we review the current state of anomaly detection in Section 2.1 and show through the literature review in Section 2.2 that there is a need of SEs in the field of anomaly detection. Sections 2.3 and 2.4 provide a background of the search and feature selection methods we use in this thesis to compute SEs. Lastly, Section 2.5 outlines important classifiers that we use in most of our work.

2.1 Anomaly Detection

Anomaly detection is the problem of identifying anomalies in a dataset, where we define anomalies as the data points that are generated by a process that is different from the process generating the normal data points. In most works, anomalies and outliers are the two most common terms used interchangeably in the context of anomaly detection to describe anomalies. However, in our work we differentiate between these two terms similarly as Emmott et al. (2015) and Siddiqui et al. (2015, 2019). In our case anomalies are data points that are generated by a process that is distinct from the process generating the normal data points, whereas, outliers are data points which deviate so much from the rest of the other data points in the dataset that they arouse suspicions that a different process generated them. We make this distinction because, in most applications, anomalies are extremely rare or unseen, therefore making it impossible to model the anomalous process. Instead of modelling the process that is generating anomalies, anomaly detection methods seek for outliers in the dataset in the hope that

they are anomalies. In this section, we review the different types of outliers and anomaly detection methods.

2.1.1 Types of outliers

2.1.1.1 Contextual outliers

A data point is a contextual outlier if it deviates significantly from the rest of the data set with respect to a specific context. The notion of context is incorporated in the structure of the dataset by two attributes called the contextual and behavioural attributes. These two attributes specify each data point in the dataset. Contextual attributes define the context of the data points. An example of contextual attributes is the longitude and latitude of a location in spatial datasets (Chandola et al. 2009) or the season of the year in time series weather data. Behavioural attributes define the characteristics of the data points. For example, in spatial datasets, this could be the average rainfall at any location (Chandola et al. 2009), whereas in time series weather data, this could be the temperature on any given day. Behavioural attributes are used to evaluate whether a data point is an outlier in the context to which it belongs. For example, in the weather time series dataset, a temperature of 5 degrees Celsius (behavioural attribute) is normal during the winter period (contextual attribute), whereas, the same temperature during a day of summer would result in an outlier due to the context. Contextual attributes already explain why a data point is outlying. Therefore, for an analyst to verify whether a detected point is anomalous, the contextual attribute can be presented to the analyst.

2.1.1.2 Collective outliers

A collection of data points are considered to be outliers if as a collection they deviate from the rest of the dataset. In this case, individual data points in the collection are not outlying and only their occurrence together as a collection is outlying (Chandola et al. 2009). Collective outliers occur in sequential, spatial and graph datasets (Chandola et al. 2009). An example of collective outliers from Sari (2015) is shown in Figure 2.1. The highlighted region shows an outlier sequence. The sequence is outlying because the sequence is flat for a long period compared to the rest of the dataset and the individual data points by themselves are not outlying (Chandola et al. 2009). The outlying collective sequence already explains to the analyst why it is outlying. It is outlying because the data points as a collection display a relationship that is different compared to the rest of the dataset.

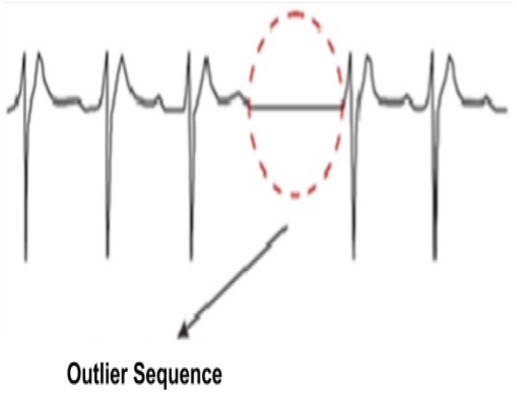


FIGURE 2.1: Collective outliers (Chandola et al. 2009)

2.1.1.3 Point outliers

A data point is a point outlier if it deviates from the rest of the datasets. Point outliers are the simplest type of outliers and are the most researched type of outliers in anomaly detection (Chandola et al. 2009). In our work, when we refer to outliers, we are referring to point outliers. For point outliers, most of the work focuses on the detection and ignores developing explanations to assist the analyst in understanding what makes it outlying. In our work, we focus on developing SEs for detected point outliers.

2.1.2 Anomaly detection methods

2.1.2.1 Supervised anomaly detection

Supervised anomaly detection methods require training data from both the anomalous and normal classes in order to build a predictive model that will learn to distinguish between anomalies and normal data points. However, anomalies are far fewer than the normal data points because they are rare and challenging to find and are unseen in most cases. In cases where a few anomalous data points are available, the predictive model will have to learn on an imbalanced dataset because the anomalies are far fewer than the normal data points. Building predictive models on imbalanced dataset involves using a cost-sensitive model or injecting the anomalous class with artificial anomalies in order to balance the two classes (Haixiang et al. 2017). Supervised anomaly detection is similar to building predictive models and is often overlooked in literature because of the lack of observed anomalies in most domains. Hence in our work, we do not consider supervised anomaly detection techniques to compute the outlier-based SEs because in real-world applications they do not have access to the model that can distinguish the processes generating the anomalous and normal data points.

2.1.2.2 Semi-supervised anomaly detection

Semi-supervised anomaly detection methods only require normal data points during training. With the normal data points, semi-supervised methods build a one class classifier that models the structure of the normal class. Examples of such models are One Class Support Vector Machines (OCSVM) (Schölkopf et al. 2001, Tax & Duin 2004) and One Class Neural Networks (Hawkins et al. 2002, Williams et al. 2002). The models are trained only using the normal class of data points. For unseen data points at test time, the one class model assigns high outlier scores to data points that deviate from the learnt decision boundary of the model. Semi-supervised anomaly detection methods are more applicable to real-world applications compared to supervised anomaly detection methods because they only require information about the normal data points. In our work, when we compute outlier-based SEs, we do not know which data points are normal beforehand. Hence we do not use semi-supervised methods to score outliers.

2.1.2.3 Unsupervised anomaly detection

Unsupervised anomaly detection methods identify anomalies in a data set without the need for labels. Unsupervised anomaly detection methods are more applicable to real-world applications because, in most application domains, normal and anomalous data points are not known beforehand. The primary assumption made by unsupervised methods is that anomalies are far fewer than the normal data points. Unsupervised anomaly detection methods operate by looking for outliers in the dataset with the hope that they are anomalies. The majority of unsupervised anomaly detection methods use either nearest-neighbour, clustering, or statistical-based methods (Chandola et al. 2009). Nearest neighbour based methods assume that normal data points lie in dense neighbourhoods, while outliers occur far from their closest neighbours. Clustering based methods assume that either normal data points belong to a cluster while outliers do not belong to any cluster or that normal data points belong to large and dense clusters, while outliers either belong to small or sparse clusters (Chandola et al. 2009). Statistical methods assume that normal data points occur in high probability regions of a stochastic model, while outliers occur in the low probability regions (Chandola et al. 2009). Most of the work in anomaly detection is unsupervised. Hence, in our work, we use unsupervised anomaly detector's outlier scoring measures to guide the search for the outlier-based SEs because they do not require labels about the data and are more applicable to real-world applications.

2.2 Literature Review

2.2.1 Anomaly detectors with explanations

There are a few anomaly detection methods that provide explicit outlier explanations that attempt to explain why the anomaly detector considered them as outliers. The idea behind these explanations is to make the analyst understand the inner workings of the anomaly detector itself by providing the analyst with outlier explanations that explain why a detected outlier point is outlying according to the anomaly detector used.

Angle-Based Outlier Detection (ABOD) (Kriegel et al. 2008) is one of the first anomaly detectors to provide outlier explanations for each detected data point. ABOD uses the variance of angles among data points in order to compute outlier scores. To calculate the outlier score of a data point, ABOD calculates the variance over the angles between the difference vectors of the data point to all the pairs of data points in the dataset weighted by the distance of the data points (Kriegel et al. 2008). The smaller the variance, the more outlying the data point is because outliers exhibit low variance of angles between pairs of data points that are further away. In order to obtain an outlier explanation, ABOD finds the closest difference data point and ranks the features in descending order according to their absolute difference. Features with a higher magnitude are considered to explain why the outlier point is outlying. This method of obtaining outlier explanations has two flaws. Firstly if the closest data point to the outlier point is also an outlier, then the features explaining the outlier data point will be incorrect. Secondly, the outlier explanation is based on the outlier data point's closest data point and disregards the rest of the data in the dataset.

Subspace Outlier Degree (SOD) (Kriegel et al. 2009) is an anomaly detection method that detects outliers in axis-parallel subspaces. For each data point, SOD analyses how well it fits in a subspace spanned by its nearest neighbours. The subspace is an axis-parallel lower dimensional hyperplane such that all the nearest neighbours are close to the hyperplane. A data point is outlying if it significantly deviates from the hyperplane in the subspace that is perpendicular to the hyperplane. To obtain an outlier explanation for a detected outlier point, they obtain the quantity defining the subspace vector perpendicular to the hyperplane for each feature.

Correlation Outlier Probability (COP) (Kriegel et al. 2012) is an anomaly detection method that assumes data points that are generated by the same process show a similar correlation among some features such that they lie on a common lower dimensional hyperplane. A data point is considered an outlier with respect to a set of normal reference data points if the outlying data point is not located on the hyperplane spanned by the

normal data points. The outlier score of a data point is calculated by first determining the hyperplane defined by the neighbours of the data point and evaluating its deviation to its neighbours in the subspace perpendicular to the hyperplane. The outlier score measures how good a data point lies on the hyperplane. The outlier explanation generated by COP consists of two main components: the error vector, which is the difference between the data point mapped back into the original feature subspace and the actual data point, and the actual outlier score, which indicates the likelihood the data point comes from a process that is different from the normal data points. Features with larger errors in the error vector are considered to be the outlying features that explain the outlier.

Local Outliers With Graph Projection (LOGP) (Dang et al. 2014) is an anomaly detection method that finds outliers using a feature transformation approach and provides an explanation in terms of the feature weights in its linear transformation. To compute the outlier score of a data point, LOGP first seeks a linear transformation that maximally separates it from its nearest neighbours in the full feature space. In the transformed space, LOGP computes the outlier score as the statistical distance from the point to its neighbours. In order to obtain the outlier explanations for a detected outlier point, the features are sorted in descending order according to the absolute value of the weight vector in the linear transformed space. Features with the largest absolute weight values are considered to be the outlying features that explain why the detected outlier point is outlying, according to LOGP.

Local Outlier Detection with Interpretation (LODI) (Dang et al. 2013) determines if a data point is an outlier by using an information theoretic measure of entropy. For each data point, LODI selects an appropriate set of neighbouring data points and then seeks an optimal one-dimensional subspace that maximally separates the point from its neighbours. Dang et al. (2013) do this by developing a method whose solution relies on matrix eigen-decomposition in order to learn the optimal one-dimensional subspace in which the data point under consideration is mostly distinguishable from its neighbouring set. The one-dimensional subspace is a linear combination of the original features and is used to obtain the features that explain why an outlier point is outlying. In order to obtain the outlier explanation for an outlier point, the features are sorted in descending order according to the absolute value of the weight vector in the linear transformed space. Features with the largest absolute weight values are considered to be the outlying features that explain why the outlier point is outlying, according to LODI.

The surveyed anomaly detectors provide explanations that attempt to explain the inner workings of the anomaly detector. Most anomaly detection methods do not provide explanations and operate as black-boxes such that it is hard to identify the features

that are responsible for a data point receiving a high outlier score. In general, obtaining outlier explanations that can explain the inner workings of each anomaly detection algorithm is a non-trivial task, and this is not the aim of our work in this thesis. The non-trivial task of explaining an anomaly detector's inner working has led to the development of outlier explanation methods that can explain any outlying data point regardless of the anomaly detector used to detect anomalies. These explanations do not explain the inner workings of the anomaly detectors but explain what makes the detected outliers outlying.

2.2.2 Outlier explanations for groups of outliers

Due to the lack of outlier explanations from anomaly detectors, there has been work developed to compute explanations for detected outliers as groups regardless of the anomaly detection algorithm used. The aim is to find an explanation about what makes the identified outliers outlying as groups, instead of explaining the outlier data points to the analyst one at a time.

Given a labelled dataset containing outlier and normal data points from an arbitrary anomaly detector, Kuo & Davidson (2016) formulate a constraints programming problem to find the optimal subset of features that differentiate the outliers from the normal data points. The constraint programming approach aims to identify an optimal feature subspace that maximises the difference between the neighbourhood densities of the outliers and normal data points. The outlier points should have fewer neighbours in this subspace, while the normal points should have many neighbours and the density of the data points with respect to the feature subspace must separate all of the outlier points from the normal data points.

Kopp et al. (2014) use Sapling Random Forests (SRF) (Pevnỳ & Kopp 2014) to first obtain the explanations of each detected outlier point. They then group the outliers using k-means clustering based on the similarities of their explanations. Outliers that land in the same cluster will share an explanation and will enable the analyst to investigate similar outliers at once instead of analysing the outliers one at a time. SRFs and their explanations are covered in more detail Section 2.2.6.

XPacs (Macha & Akoglu 2018) aims to explain known anomalies in groups. It takes as input labelled anomalous and normal data points. XPacks uses clustering to find small micro-clusters of anomalies hidden in arbitrary feature subspaces that collectively and yet succinctly represent the anomalies and separate them from the normal points (Macha & Akoglu 2018). Features that fall part of the feature subspace in which micro-clusters lie form the clusters explanation. XPacks explanations are rule-based explanations that

constrain the features between a range of values which describe what makes an anomaly fall part of the cluster.

Angiulli et al. (2013) introduce the EXPREX algorithm that aims to explain groups of outlier data points as one cluster. EXPREX searches for `<explanation,feature>` pairs, where the single feature differentiates as many outliers as possible from the normal data points that share the same context as the explanation. The explanations are rule-based explanations that satisfy a set of conditions whose aim is to differentiate as many outliers as possible from the normal data points.

Explaining outliers in a group or a set of groups has two key advantages (Macha & Akoglu 2018). Firstly, it saves the analyst time, rather than the analyst having to go through the outliers one at a time. Secondly, it draws attention to outliers that form patterns, which are potentially more critical because they are repetitive. In our work, we are more interested in developing methods that can explain detected outliers one at a time rather than explaining them in groups. We are interested in explaining outliers one at a time to the analyst because firstly, outliers do not always fall into clusters or groups because they might be scattered. Secondly, the clusters could miss important features for certain outliers, which will be critical to the analyst's judgment. Lastly, the field of explaining outliers to analysts one at a time is still novel, and SEs are required to cover the gaps we have identified in our problem statement.

2.2.3 Outlying aspects mining

Outlying aspects mining (Duan et al. 2015, Vinh et al. 2015, 2016) identify which subset of features make a data point $\mathbf{x}$ different from the rest of the dataset. It is used to obtain the best feature subspace that distinguishes a data point from the rest of the dataset, whether it is an outlier or a normal data point. For example, in soccer, a coach may want to know the most distinguishing features of each of his players in order to know their strengths and weaknesses within the team. Outlying aspects mining tackles the problem from three different angles known as the score-and-search, feature selection and hybrid-based methods.

2.2.3.1 Score-and-search based methods

Let $\mathbf{x}$ represent the data point whose outlying aspect need to be computed and $\mathbf{x}_{\mathcal{S}}$ denote the projection of the data point $\mathbf{x}$ onto the arbitrary feature subspace specified by $\mathcal{S}$. Score-and-search based methods first define an outlier scoring measure $\phi : \mathbb{R}^{|\mathcal{S}|} \longrightarrow \mathbb{R}$ to measure how outlying $\mathbf{x}$ is compared to the rest of the dataset in feature subspace $\mathcal{S}$.

The outlying aspects of $\mathbf{x}$ are obtained by computing the outlier scoring measure of all possible feature subspaces $\mathcal{S}$ and selecting the subspace that returns the best score as the outlying aspects of $\mathbf{x}$.

Zhang et al. (2004) are the first to define the outlying aspect mining problem. They introduced HOS-Miner, which employs a distance-based outlier scoring measure that is the sum of distances between the data point of interest $\mathbf{x}$ and its k-nearest neighbours (k-nn).

$$\phi(\mathbf{x}_{\mathcal{S}}) = \sum_{\mathbf{z} \in k\text{-nn}(\mathbf{x}_{\mathcal{S}})} d(\mathbf{x}_S, \mathbf{z}), \tag{2.1}$$

where $d(\mathbf{x}_S, \mathbf{z})$ is the Euclidean distance between the two data points in subspace $\mathcal{S}$, k-nn$(\mathbf{x}_{\mathcal{S}})$ is the set of k-nearest neighbours of $\mathbf{x}_{\mathcal{S}}$. HOS-Miner then defines a distance threshold of δ and uses the monotonicity property of the outlier measure to prune the search space in order to avoid an exhaustive search through the feature space. Distance-based measures monotonically increase with the number of features. If $\phi(\mathbf{x}_{\mathcal{S}}) \geq \delta$ then $\mathbf{x}$ will remain outlying in any subspace that is a superset of $\mathcal{S}$, therefore, all the subspaces that are supersets of $\mathcal{S}$ can be excluded from the search. Also, if $\phi(\mathbf{x}_{\mathcal{S}}) < \delta$ then $\mathbf{x}$ will not be outlying in any subspace that is a subset of $\mathcal{S}$. Therefore, all the subspaces that are subsets of $\mathcal{S}$ can be excluded from the search. The final step in HOS-Miner is to select from the remaining feature subspaces, the feature subspace with the largest outlier score as the most outlying. The problem with the final step is that they compare the outlier scoring measures of different feature subspaces of different sizes. Comparing outlier scoring measure of feature subspaces of different sizes is a problem because there is a bias towards subspaces of higher dimensionality.

In order to address the dimensionality bias in HOS-Miner, Duan et al. (2015) introduced OAMiner. OAMiner uses the kernel density estimation measure (Silverman 1986) as their outlier scoring measure. Duan et al. (2015) found that the density of the outlier scoring measure decreases as dimensionality increases. The dimensionality bias is due to the distance-based calculation of the outlier scoring measure. To avoid dimensionality bias Duan et al. (2015) introduce the density rank to obtain the outlying aspects. The density rank works as follows: the density of every data point in the dataset needs to be computed in each feature subspace; then in each feature subspace, the ranking of every data point according to its density is tabulated in descending order; for a data point $\mathbf{x}$, the feature subspace that obtains the highest rank is its outlying aspect

Vinh et al. (2016) identified two problems with OAMiner. First, the density rank is a very computationally expensive measure because they need to compute the density

of every data point in every feature subspace. OAMiner introduced an upper bound to prune the search space and limited the total number of features in the outlying aspect subspace to a maximum of 5. Even with the pruning, scaling OAMiner to large data sets is a challenging and computationally expensive task. The second problem is that even though ranking provides a normalised measure for comparing densities across feature subspaces of different size, important information regarding the absolute degree of deviation is lost. For example in Figure 2.2 (a), the data point (red square) is ranked first in feature subspace $\{D_1, D_2\}$ according to the density rank, whereas, in Figure 2.2 (b) the feature subspace $\{D_3, D_4\}$ is ranked fifth (Vinh et al. 2016). Based on OAMiner, the feature subspace $\{D_1, D_2\}$ is the outlying aspect for the data point; however, it is clear that feature subspace $\{D_3, D_4\}$ is where the data point deviates the most from the rest of the dataset (Vinh et al. 2016).

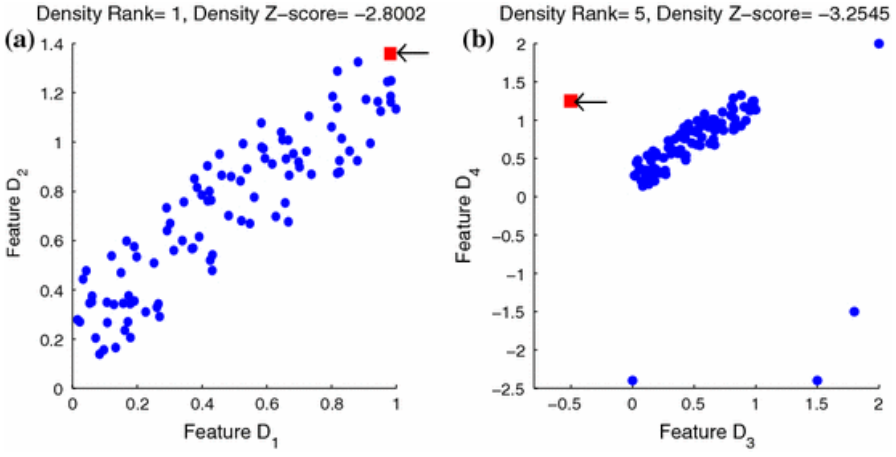


FIGURE 2.2: OAMiner and the density Z-score outlying aspects of the data point in red (Vinh et al. 2016). Feature subspace $\{D_3, D4\}$ is where the data point deviates the most from the rest of the data set. However, OAMiner identified the feature subspace $\{D_1, D_2\}$ as the outlying aspects based on the density ranking. The density Z-score correctly identifies feature subspace $\{D_3, D4\}$ as the outlying aspects of the data point.

Vinh et al. (2016) address the problems of OAMiner by developing an alternative to the density rank and identify two outlier scoring measures that are not bias in terms of dimensionality. They introduce a new measure called the density Z-score:

$$Z(\phi(\mathbf{x}_{\mathcal{S}})) = \frac{\phi(\mathbf{x}_{\mathcal{S}}) - \mu_{\phi_{\mathcal{S}}}}{\sigma_{\phi_{\mathcal{S}}}}, \tag{2.2}$$

where $\mu_{\phi_{\mathcal{S}}}$ and $\sigma_{\phi_{\mathcal{S}}}$ are the mean and standard deviation of the density of all the data points in subspace $\mathcal{S}$. In Figure 2.2, the density Z-score correctly selects feature subspace $\{D_3, D4\}$ as the outlying aspects. The density Z-score retains information regarding the absolute degree of deviation, unlike the density rank. Similar to the density rank, the density Z-score is a computationally expensive measure to compute because it requires

the computation of the density of every data point in every feature subspace in order to find a data points outlying aspects. Vinh et al. (2016) used a beam search to prune the feature subspace to limit the total number of features in the outlying aspects to a maximum of 3. In addition to the density Z-score, Vinh et al. (2016) found that the Local Outlier Factor (LOF) and Isolation Forests (iTree) anomaly detectors have outlier scoring measures that are not dimensionality bias. Therefore the LOF and iTree outlier scoring measures do not need to be normalised because their scoring measures are not biased towards higher dimensional feature subspaces.

2.2.3.2 Feature selection based methods

Micenková et al. (2013) developed the first outlying aspect mining feature selection based method. The main objective is to transform the outlying aspect mining problem into a classic two class feature selection problem. To find the outlying aspects of a data point $\mathbf{x}$, Micenková et al. (2013) first construct a synthetic class by randomly drawing $2 \cdot k$ samples from a standard normal Gaussian distribution, where the mean $\mu = \mathbf{x}$ and the variance is chosen such that the samples generated closely resemble $\mathbf{x}$. The Gaussian distribution generates samples that are similar to $\mathbf{x}$. The second class representing the rest of the data is constructed by first selecting the k-nearest neighbours of $\mathbf{x}$ in the full feature space and then sample k data points from the rest of the dataset while excluding the k-nearest neighbours of $\mathbf{x}$ and $\mathbf{x}$ itself. The first class contains $2 \cdot k$ data points that are similar to $\mathbf{x}$ is the outlier class of $\mathbf{x}$, whereas the second class also containing $2 \cdot k$ data points represent the rest of the dataset is the inlier class of $\mathbf{x}$. To obtain the outlying aspects of $\mathbf{x}$ any feature selection algorithm can be used to select the best features that separate the two classes.

2.2.3.3 Hybrid methods

Vinh et al. (2015) introduced a hybrid approach that tackles the outlying aspect mining problem through two stages. The first stage ranks the features according to their potential to make $\mathbf{x}$ outlying, while the second stage is a search-and-score based approach on the top-ranked features from the first stage. Vinh et al. (2016) formulated the first stage into a quadratic programming problem in order to find the top features that make $\mathbf{x}$ outlying. In the second stage Vinh et al. (2015) used the density rank measure to obtain the most outlying features from the top features obtained from the first stage. The hybrid approach gives the added advantaged of pruning the features in the first stage so that there are fewer features for the score-and-search stage.

Outlying aspect mining and our work are very similar with a few important differences to note. In our work, we are not interested in finding the most outlying feature subspace. In order to find the most outlying feature subspace the computationally expensive density Z-score or density rank need to be used in order to avoid dimensionality bias. To compute outlier-based SEs, we are not comparing outlier scoring measures of feature subspaces of different sizes because we are only interested in identifying the best feature subspace for each explanation of size i in the SE. SEs are therefore computationally cheaper than outlying aspect methods because we do not have to use density Z-score or density rank because we only compare feature subspaces of the same size against each other.

2.2.4 Outlying property detection

Given a data point $\mathbf{x}$, outlying property detection (Angiulli et al. 2017, 2009) finds the disjoint feature subspaces a and E such that $\mathbf{x}$ is outlying in the feature a with respect to the data points it is mostly similar to in E. In this case, E is the explanation that explains why $\mathbf{x}$ is outlying in the feature a compared to its most similar data points. The explanation E is a contextual rule-based explanation that satisfies certain rule-based conditions. For an analyst to investigate the outlying property of $\mathbf{x}$, they first need to select the data points in the feature subspace E meeting the conditions specified by the rules of E, then view $\mathbf{x}$ in the outlying feature a. The feature a is called the outlying property.

The explanations in outlying property detection and our work are vastly different. Outlying property detection identifies the feature a that makes a data point different from the data points that are most similar to it. Whereas, in our work, we are interested in identifying the subsets features that explain why a data point is outlying.

2.2.5 Pictorial Explanations

Gupta et al. (2018) introduce LookOut which explains outliers through what they call focus plots. Given outliers from an arbitrary anomaly detector, LookOut finds two-dimensional scatter plots that visualise the feature subspace where the outliers are deviating from the rest of the dataset. With visualisations of the outliers, the analyst can then quickly interpret and spot the outliers and verify if they are anomalous or not given the visualisations of feature pairs. LookOut considers $\binom{d}{2}$ features subspaces by generating plots of all feature pair combinations, where d is the total number of features in the full feature space. Within each plot, LookOut scores the outlyingness of all of the

detected outliers from the anomaly detector. In order to limit the number of plots presented to the analyst, LookOut finds the best plots where the outliers have the highest outlying scores.

In our work, we are interested in identifying feature subspaces that are beyond two-dimensional. Also, LookOut visualises the outliers as groups while we are more interested in explaining outliers one at a time.

2.2.6 Explaining outliers using decision rules

Given an outlier from any arbitrary anomaly detector, Explainer (Pevnỳ & Kopp 2014) returns explanations explaining what makes it outlying in disjunctive normal form (DNF). If $\mathbf{x}$ is outlying, then its explanation E is of the form:

$$E = (\mathbf{x}_{j_1} > \theta_1) \wedge (\mathbf{x}_{j_2} < \theta_2) \wedge \cdots \wedge (\mathbf{x}_{j_t} > \theta_t). \tag{2.3}$$

The explanation E is read as: $\mathbf{x}$ is an outlier because feature j_1 is greater than θ_1 and feature j_2 is smaller than θ_2 and so on. Explainer uses SRFs to learn the explanation E from training multiple decision trees. For each detected outlier, Explainer presents features together with rules on them, describing why it is outlying. To obtain E, Explainer trains many binary decision trees using CART (Lewis 2000) to separate $\mathbf{x}$ from the rest of the dataset. Rules along the path from the root node to the leaf are extracted and returned in DNF as the explanation E. To construct the training set, Explainer constructs two classes. The first class contains only the outlying data point $\mathbf{x}$, and the second class consists of k randomly selected data points from the rest of the data. The imbalance of the two classes causes the decision trees that are trained to be of very low in height, typically between 1 and 3, resulting in very short explanations. Explainer trains multiple decision trees from random samples of the dataset to obtain multiple explanations which it aggregates in the end.

Explainer's explanations are very short and usually contain between 1 to 3 features. In our work, we are interested in computing as many features in the explanation as required by the analyst. Also, a problem we observed is the significant imbalance used to train the decision trees to obtain the explanations.

2.2.7 Sequential Feature Explanations

Siddiqui et al. (2019, 2015) introduced an outlier explanation called an SFE that will work with any density based anomaly detector, provided that the anomaly detector can

return an outlier score for a point in any given feature subspace. A length k SFE for a data point $\mathbf{x}$ is an ordered list of feature indices $\mathcal{E} = (e_1, \ldots, e_k)$ where $k < d$ and d is the dimensionality of the data point $\mathbf{x}$. Features that appear earlier in the order are considered to be more important to the high outlier score of a point. The notation $\mathcal{E}_i$ denotes the set of the first i feature indices of $\mathcal{E}$. Also, for any set of feature indices $\mathcal{S}$ and a data point $\mathbf{x}$, $\mathbf{x}_\mathcal{S}$ denotes the projection of $\mathbf{x}_\mathcal{S}$ onto the feature subspace specified by $\mathcal{S}$. Therefore, given an SFE $\mathcal{E}$ for a data point $\mathbf{x}$, the point is incrementally presented to the analyst by first presenting feature $\mathbf{x}_{\mathcal{E}_1}$. If the analyst can make a judgement based on only that information, then the analyst's job is done. Otherwise, the analyst is shown the next feature together with the previous feature, which is $\mathbf{x}_{\mathcal{E}_2}$. The process of incrementally adding features to the set of presented features continues until the analyst can make a decision. Siddiqui et al. (2019, 2015) developed the Marginal and Dropout SFE methods to explain to the analyst why an outlier point is outlying according to the anomaly detector and for the analyst also to understand the inner working of the anomaly detector. Let f be a density-based anomaly detector that predicts the probability of a data point being normal in any given feature subspace. There are two Marginal SFEs, namely the sequential marginal SFE (SeqMArg) and the independent marginal SFE (IndMarg). SeqMarg adds one feature at a time to the SFE $\mathcal{E}$. At each step, it adds the feature that minimises the joint marginal density with the previously selected features in the SFE. More formally, SeqMarg computes the following explanation:

$$\textbf{SeqMarg}: \qquad e_i = \arg\min_{j \in \overline{\mathcal{E}}_{i-1}} f(\mathbf{x}_{\mathcal{E}_{i-1}+j}), \tag{2.4}$$

where $\overline{\mathcal{E}_i}$ is the complement and represents the set the feature indices not selected in $\mathcal{E}_i$. SeqMarg takes into account feature interactions because it considers previously selected features in the SFE before it adds a new feature. IndMarg is a marginal method that does not take feature interactions into accounts and assumes features are independent. IndMarg only requires computation of individual marginals $f(\mathbf{x}_i)$ and adds features into the SFE $\mathcal{E}$ by sorting the features in increasing order of $f(\mathbf{x}_i)$. There are two Dropout SFEs, namely the sequential dropout SFE (SeqDO) and the independent dropout SFE (IndDO). IndDO assigns a score of $f(\mathbf{x} - \mathbf{x}_i) - f(\mathbf{x})$ for each feature i, where we denote the removal of $\mathbf{x}_i$ from $\mathbf{x}$ by $\mathbf{x} - \mathbf{x}_i$. To obtain the IndDo SFE, the features are sorted in descending order of their scores. SeqDo is defined as:

$$\textbf{SeqDO}: \quad e_i = \arg\max_{j \in \overline{\mathcal{E}}_{1:i-1}} f(\mathbf{x}_{\overline{\mathcal{E}}_i - j}). \tag{2.5}$$

2.3 Search methods

Computing the outlier-based SEs and some of the sample-based SEs requires a combinatorial search method in the feature space to guide the search of the subset of features to add into the SEs. In this section, we conduct a brief survey on search methods available in the literature. We broadly classify the search methods into the sequential selection and meta-heuristic search algorithms.

2.3.1 Sequential search algorithms

There are two types of sequential search algorithms, the Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) sequential search methods. SFS initially starts with an empty set and adds the feature in the first step that returns the highest value for the objective function. In the second step, it adds the feature that returns the highest value of the objective function together with the first feature. The process of continually adding features that maximise the value of the objective function, along with the already chosen features continues until we obtain the required number of features. SBS starts with a complete set of features and removes one feature at a time whose removal gives the lowest decrease in the objective function. SFS and SBS are both greedy methods because their outputs are both nested. To compute feature subsets that are not nested the SFS and SBS algorithms were modified to create the Sequential Forward Floating Selection, Sequential Backward Floating Selection, Adaptive Sequential Forward Floating Selection, Adaptive Sequential Backward Floating Selection, Plus-L-Minus-r and Improved Forward Floating Selection (Nakariyakul & Casasent 2009, Somol et al. 1999) and search algorithms. However, sequential methods are exploitative methods that tend to get stuck in sub-optimal local minima solutions with no means of escape.

2.3.2 Meta-heuristic search algorithms

Meta-heuristic search algorithms are search methods that trade-off between randomisation and local search. Randomisation is incorporated to keep a balance between exploration and exploitation. Exploration enables the meta-heuristic to escape local minima by using randomisation techniques to allow the search to spread to new solutions that are far from the current area of search such that all regions in the search space are visited at least once (Hussain et al. 2018). On the other hand, exploitation uses past search experience by using a greedy local search that tries to quickly converge to an optimal solution without wasting too many moves (Hussain et al. 2018). Meta-heuristic search

algorithms are categorised as either single or population-based searches. Single-based searches focus on modifying and improving a single candidate solution. Single solution meta-heuristics include simulated annealing (Van Laarhoven & Aarts 1987), iterated local search (Lourenço et al. 2003), variable neighbourhood search (Mladenović & Hansen 1997), guided local search (Voudouris & Tsang 2003) and greedy randomised adaptive search procedure (Resende 2009). Population-based approaches maintain and improve multiple candidate solutions. A population of multiple solutions is generated initially; then, in every iteration, a set of solutions are manipulated to find solutions toward better search areas. Population-based meta-heuristics include genetic algorithms (Whitley 1994), and particle swarm optimisation (Shi et al. 2001) and ant colony optimisation (Dorigo & Di Caro 1999). Single solution based meta-heuristics are a bit more exploitation based and are classified as local search optimisation methods because they have a lower chance of finding global optima. Population-based meta-heuristics are more explorative based and are known as global search optimisation methods because of their explorative nature that gives them a higher chance of obtaining the global optima.

The SFE methods used sequential algorithms to compute the SFEs. In our work, we also use sequential algorithms and introduce a novel PSO meta-heuristic method to solve the nesting problem. We chose the PSO because it is an explorative global search optimisation method that has been tried and tested in literature (Kennedy 2010, Shi et al. 2001).

2.4 Feature Selection

For the sample-based SEs, we will use various feature selection methods to compute the SEs. In this section, we detail the different types of feature selection methods available in the literature.

Feature selection eliminates irrelevant features in order to obtain a richer feature subset of relevant features that will improve the performance of predictors, reduce computational requirements, reduce the effect of the curse of dimensionality and improve interoperability of the data from the users perspective. Feature selection methods fall into one of the following categories: filter, wrapper and embedded-based methods.

2.4.1 Filter based methods

Filter based methods make use of a feature ranking function to measure the relevance of each feature. Features with the highest relevance measure are the chosen features

to represent the data best. This method is the cheapest computationally compared to the other feature selection methods; however, it does not take feature relationships into account because it assesses each feature independently. Some of the most used filter measures are the Chi-squared test, mutual information and correlation coefficient measures (Novaković 2016).

2.4.2 Wrapper based methods

Wrapper based methods obtain a feature subset by using a classifier as a black-box and the classifier's performance as the objective function guided by a search algorithm to identify the best features that represent the data (Chandrashekar & Sahin 2014). The wrapper based methods search for the best feature subsets using either sequential or meta-heuristic search algorithms along with a classifier that can model the data well.

2.4.3 Embedded based methods

Embedded based methods perform feature selection during the training process of the classifier (Saeys et al. 2007, Chandrashekar & Sahin 2014). Embedded based methods use the coefficients associated with a feature in the trained model to measure its relevance, and the top-ranked features are selected. However, this results in features to be assessed independently without taking feature relationships into account. To overcome this, Guyon et al. (2002) introduced Recursive Feature Elimination (RFE) , which works as follows:

1. Train the classifier to optimise its coefficients on the features;
2. Rank all the feature;
3. Remove the feature with the lowest rank.

This iterative process continues until there are no more features to train the model. The next step is to rank features in descending to the order of their removal. The best features are the features with the highest ranks.

2.5 Classifiers

For the sample-based SEs and the evaluation of our SEs, we need to use classifiers that will compute the SEs and train a simulated analyst, respectively. We chose the Support

Vector Machine (SVM) (Schölkopf et al. 2001, Hsu et al. 2003) and Random Forest (RF) (Breiman 2001) classifiers in our work because they fall part of the state of the art classifiers in machine learning and have been used extensively throughout literature to solve complex problems.

2.5.1 Support Vector Machines

SVMs (Schölkopf et al. 2001, Hsu et al. 2003) are powerful non-linear models used in classification and regression problems. Assuming we have N data points $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}$, where each $\mathbf{x}^{(i)}$ is a d-dimensional real valued-vector. Let $y^{(i)} \in \{-1, 1\}$ represent the classes that each $\mathbf{x}^{(i)}$ belongs to. The main objective of an SVM is to construct a hyperplane in high dimensional spaces that linearly separate the data points with respect to the classes. It achieves this by constructing a hyperplane that has the largest distance to the nearest training data points of the two classes. If the original problem which lies in $\mathbb{R}^d$ space is not linearly separable in that space, then the original finite-dimensional space will be mapped into a higher-dimensional space (eg. $\mathbb{R}^d \longrightarrow \mathbb{R}^p$, where $p > d$), where it can be linearly separated.

More formally, the SVM problem is the following optimisation problem:

$$\arg\min_{\mathbf{w},\xi,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \right\} \tag{2.6}$$

subject to

$$y^{(i)}(\mathbf{w} \cdot \psi(\mathbf{x}^{(i)}) + b) \geqslant 1 - \xi_i \tag{2.7}$$

where $\mathbf{w}$ is the normal vector to the plane that separates the two classes, b is a constant, $\psi : \mathbb{R}^d \longrightarrow \mathbb{R}^p$ is the mapping function, ξ_i are the non-negative slack variables, which measures the degree of misclassification of the data point $\mathbf{x}^{(i)}$ and C is the regularisation parameter.

This leads to the following dual formulation (Schölkopf et al. 2001, Hsu et al. 2003):

$$\min_{\alpha} \frac{1}{2}\sum_{\forall i,j} \alpha_i \alpha_j \psi(\mathbf{x}^{(i)}) \cdot \psi(\mathbf{x}^{(j)}) - \sum_i y^{(i)}\alpha_i \tag{2.8}$$

subject to $\sum_{\forall i} \alpha_i y^{(i)} = 0$ and $0 \leqslant \alpha_i \leqslant C, \forall i$

Where the α_i variables are defined such that:

$$\mathbf{w} = \sum_{\forall i} \alpha_i y^{(i)} \psi(\mathbf{x}^{(i)}) \tag{2.9}$$

The decision rule then becomes:

$$f(\mathbf{x}) = \sum_{\forall i} y^{(i)} \alpha_i \psi(\mathbf{x}^{(i)}) \cdot \psi(\mathbf{x}) + b = \sum_{\forall i} y^{(i)} \alpha_i K(\mathbf{x}^{(i)}, \mathbf{x}) + b \tag{2.10}$$

where $K(\mathbf{x}^{(i)}, \mathbf{x}) = \psi(\mathbf{x}^{(i)}) \cdot \psi(\mathbf{x})$ is known as the kernel function.

Mostly used kernel functions are (Schölkopf et al. 2001, Hsu et al. 2003):

- Linear: $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}$
- Gaussian radial basis function: $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp(-\gamma \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2)$ for, $\gamma > 0$
- Polynomial : $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)} + r)^d$
- Hyperbolic tangent: $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \tanh(\kappa \mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)} + c)$, for some $\kappa > 0$ and $c < 0$

The parameters of the kernels and the regularisation parameter C can be determined by using the grid search method (Bergstra & Bengio 2012) and k-fold cross-validation to measure the performance of the chosen parameters (Kohavi et al. 1995).

2.5.2 Random Forests

Decision trees (Quinlan 1986) are very popular statistical and machine learning models for classification and regression problems, however, they are known to be grown very deep which results in them overfitting the training data and not returning good results for unseen data (Liaw & Wiener 2002). A RF (Breiman 2001) is an ensemble model that builds multiple decision trees and merges them to reduce overfitting and get more accurate and stable predictions. RFs constructs each tree as follows:

1. Each tree randomly samples roughly two-thirds of the original dataset with replacement as its training set.
2. Each node in each tree is split using the best subset of features from a subset of randomly chosen features, whereas in decision trees, nodes are split by examining every feature and picking the best split from all the features. In our work, the best split is chosen based on the Gini Impurity (Breiman 2001) method.

The training error of the RF is estimated using the out of bag (OOB) error rate. The calculation of the OOB rate is the average of the misclassification rate on the one-third of the data omitted in each tree's training set. The majority vote aggregated from each tree is used to predict a new data point's class at test time.

Chapter 3

Research Methodology

The purpose of our study is to develop methods of computing SEs and compare them based on their performance and complexity. In our methodology, we developed two methods of computing SEs called the outlier and sampled-based SEs. Through our methodology, we do the following: analyse the performances and complexities of different anomaly detector's outlier scoring measures and search methods used to guide outlier-based SEs; analyse the performances and complexities of different feature selection methods of the sample-based SEs; compare and contrast the outlier and sample-based SEs; address the problems in the evaluation of SEs.

The rest of this chapter is outlined as follows: Section 3.1 provides details of the research design , Section 3.2 covers the data collection and processing of the anomaly benchmark datasets, Section 3.3 presents notation we will use throughout the thesis, Section 3.4 presents the outlier-based SEs , Section 3.5 presents the sample-based SEs and Section 3.6 presents the performance evaluation method we will use to evaluate our SEs.

3.1 Research Design

Figure 3.1 presents the research design of this chapter. The research design consists of four phases: Data Collection and Data Processing, Sequential Explanations which consists of the sampling and outlier-based SEs, Hyperparameter Optimisation and lastly, the Performance Evaluation of the SEs.

The Data Collection and Data Processing stage involves how the anomaly benchmark datasets we used were collected and how we normalise the data so that the features lie on a common scale. The Sequential Explanations stage involves the computation of the

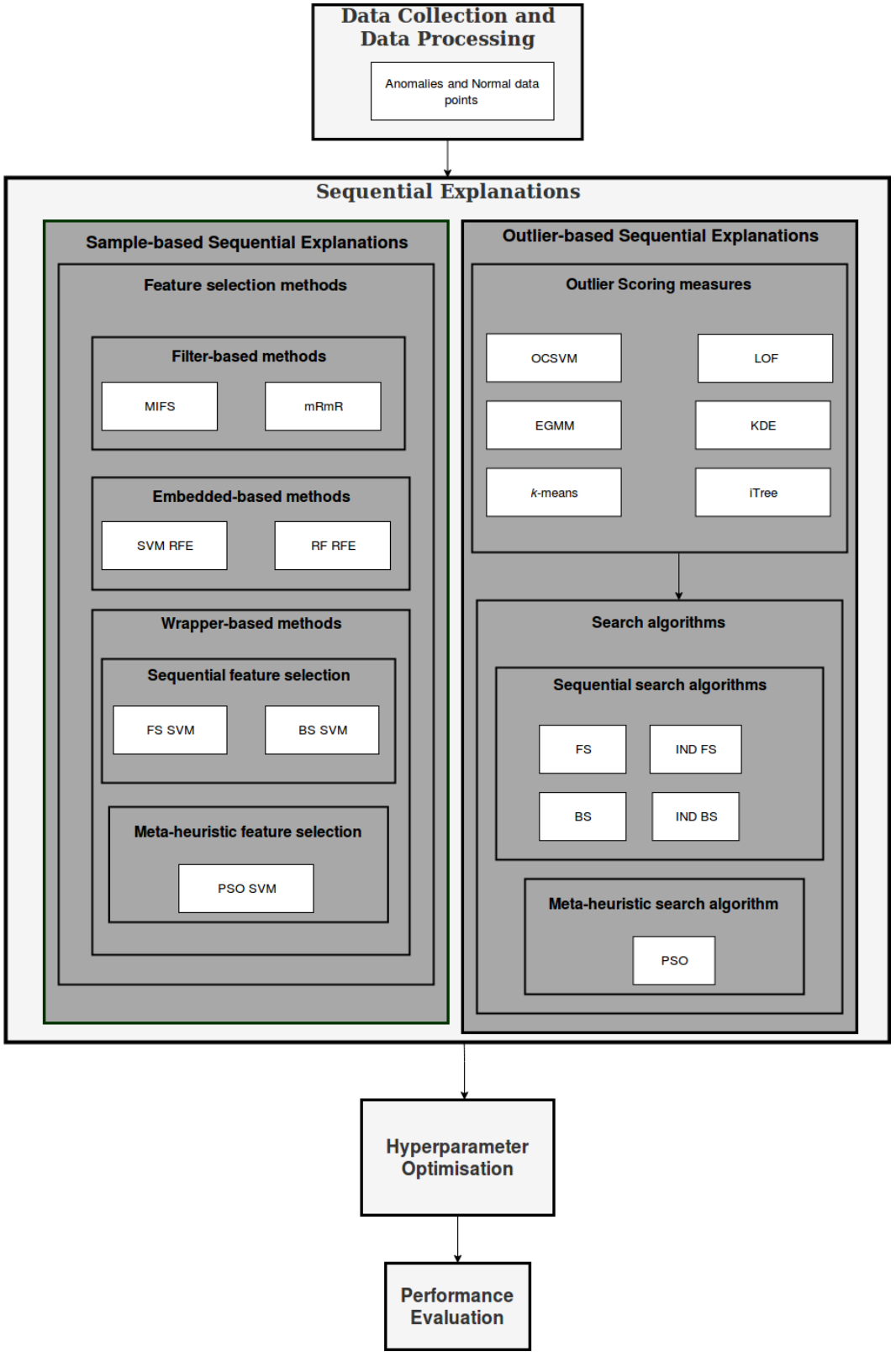


FIGURE 3.1: Research Design

SEs for the sample and outlier-based methods. For the outlier-based SEs, we compute a data point's SEs using different outlier scoring measures guided by a search algorithm in order to obtain the features that maximise the outlyingness of the data point. For the sample-based SEs, we first sample the nearest neighbours of the data point whose SEs we want to compute and then create an outlier and inlier class for that data point in order to turn the problem into a classical feature selection problem and then use feature selection methods to find the best features that separate the two classes. The Hyperparameter Optimisation stage outlines how we will tune the parameters of the outlier and sample-based SEs on the anomaly benchmark datasets. Lastly, the Performance Evaluation stage outlines how we evaluate the performance of the SEs.

For each anomaly data point from an anomaly benchmark dataset, we follow the research design presented in Figure 3.1 to compute its SEs and to answer our research questions.

3.2 Data Collection and Data Processing

3.2.1 Data Collection

There is a lack of publicly available real-world labelled anomaly detection benchmark dataset in the research community (Goldstein & Uchida 2016, Emmott et al. 2015). Therefore, in order for the research community to have anomaly benchmark datasets to benchmark their algorithms on, they usually generate their anomaly benchmark datasets by using classification datasets available from the UCI machine learning repository (Dua & Karra Taniskidou 2017) to generate anomaly benchmark datasets (Goldstein & Uchida 2016, Emmott et al. 2015). In most instances, anomaly detection benchmark datasets are created from supervised learning labelled datasets by selecting one class as the anomalous class and randomly sampling a small portion of the data points to create it, whereas, the normal class is created by randomly sampling a larger portion of the data points from another class (Goldstein & Uchida 2016). However, despite this, they still fail to make their data publicly available and thus make it difficult to reproduce their results. Emmott et al. (2015) and Goldstein & Uchida (2016) outlined methods of turning classification datasets into anomaly benchmark datasets and made them publicly available for other researchers to use to benchmark their anomaly detection algorithms. Sections 3.2.1.1 and 3.2.1.2 describe the selected anomaly benchmarks datasets from Goldstein & Uchida (2016) and Emmott et al. (2015) respectively. We use these datasets in our experiments. Tables 3.1 and 3.2 contain summarised information of the datasets.

TABLE 3.1: Goldstein & Uchida (2016) anomaly detection benchmark datasets summary

Data set	Size	#Features	#Anomalies
ALOI	50000	27	1508
Annthyroid	6916	21	250
Satellite	5100	36	75
KDD99	620098	38	1052

3.2.1.1 Goldstein & Uchida (2016) anomaly benchmarks

The following anomaly detection benchmark datasets were created by Goldstein & Uchida (2016). We briefly describe each dataset and how they turned it into an anomaly benchmark dataset. Table 3.1 provides a summary of the datasets from Goldstein & Uchida (2016).

Amsterdam Library of Object Images: The original Amsterdam Library of Object Images (ALOI) dataset (Geusebroek et al. 2005) contains about 1100 images of 1000 small objects taken under different light conditions and viewing angles. Feature vectors containing 27 features were extracted from the images using HSB color histograms. Goldstein & Uchida (2016) chose a portion of the images as anomalies in order to create the anomaly benchmark dataset.

Annthryoid: The Annthryoid dataset (Dua & Karra Taniskidou 2017) aims to determine whether a patient referred to the clinic is hypothyroid. The normal class contains patients that are not hypothyroid, while Goldstein & Uchida (2016) sampled the anomalous class from patients who are hyperfunction or subnormal functioning.

Satellite: The Satellite dataset (Dua & Karra Taniskidou 2017) comprises of features extracted from satellite observations, where they take four different light wavelengths for each satellite image. The original datasets objective is to classify the image into the soil category of the observed region. In our case, the classes "red soil", "grey soil", "damp grey soil" and "very damp grey soil" formed part of the normal class. While the classes "cotton crop" and "soil with vegetation stubble" were randomly sampled to form part of the anomaly class.

KDD99: The dataset contains simulated normal and attack traffic on an IP level in a computer network environment. In the past, anomaly detection benchmarks were created by just randomly sampling from the attacks, but due to the nature of some attacks,

TABLE 3.2: Emmott et al. (2015) motherset anomaly detection benchmark datasets summary

Motherset	#Features	#Anomaly class size	#Normal class size	#Anomaly Benchmarks	#Data Points (Range)
Pageb	10	560	4913	129	195-4466
Shuttle	9	9074	48926	120	6000-6000
Abalone	7	2081	2096	295	885-1906
Concrete	8	515	515	180	249-468
Wine	11	4113	4113	180	3165-3739
Yeast	8	977	977	120	400-888
Magic Gamma	10	12332	12332	300	2000-6000

it did not always represent a point anomaly detection problem. Therefore, Goldstein & Uchida (2016) only consider HTTP traffic. In the end, the anomaly benchmark dataset contains 620089 instances, with only 0.17% anomalies. According to Goldstein & Uchida (2016), this is the largest dataset in terms of the number of data points used so far for unsupervised anomaly detection.

3.2.1.2 Emmott et al. (2015) anomaly benchmarks

Emmott et al. (2015) are more methodological in creating their anomaly benchmark datasets, whereas, Goldstein & Uchida (2016) randomly sampled data points in order to create anomaly benchmark datasets. Emmott et al. (2015) produced anomaly benchmark datasets that vary across several dimensions that they deemed essential to real-world applications. These dimensions were point difficulty, the relative frequency of anomalies, clusteredness of anomalies, and the relevance of features. Point difficulty is a measure of the similarity of the anomalous data points to the normal ones. The higher the point difficulty of an anomalous data point, the harder it is to distinguish it from a normal data point. Relative frequency measures the frequency of anomalies in the dataset, relevance of features controls how many irrelevant features are present in the dataset. While clusteredness of anomalies measures how clustered or scattered the anomalies are. Emmott et al. (2015) constructed datasets from supervised classification and regression datasets. For regression datasets, they computed the median of the regression response and partitioned the data into two classes by thresholding on the median. The low and high values are assumed to correspond to different generative processes, while data points towards the median will have high point difficulty. For multiclass datasets, they partitioned the available classes into two sets to maximise the difficulty of telling them apart instead of selecting one class as the anomalous class and the rest as the normal class as done by Goldstein & Uchida (2016).

Emmott et al. (2015) named the original datasets that they were sampling from to create their anomaly benchmarks datasets the "motherset" datasets. For each motherset, they created an anomalous and normal class. Then to create their corpus of anomaly detection benchmark datasets, they sampled data points from the motherset one at a time such that they obtain desired properties in the anomaly benchmarks. In the end, they created thousands of anomaly benchmark datasets each with its own set of desired properties from each motherset. We selected a few of their motherset datasets to use in our experiments. We concentrated on anomaly benchmark datasets that have an anomaly frequency between $1\% - 2\%$. Table 3.2 gives a summary of the motherset datasets we used in our experiments. For example, the Abalone motherset was used to generate 295 anomaly benchmarks with an anomaly frequency between $1\% - 2\%$. The motherset contains 2081 and 2096 data points in its anomaly and normal classes, respectively. While the number of data points in the anomaly benchmark datasets created from the motherset range between $885 - 1906$. Below are brief descriptions of the selected motherset datasets (Dua & Karra Taniskidou 2017):

Pageb: A classification dataset that consists of blocks from a page layout of a document. The aim is to classify each block that has been detected by a segmentation process.

Shuttle: A multiclass classification dataset that describes radiator positions in a NASA space shuttle.

Abalone: A regression dataset for predicting the age of abalone from physical measurements.

Concrete: A regression dataset for predicting the compressive strength in concrete.

Wine: A multiclass classification dataset containing chemical measurements of wine. The chemical measurements are used to predict the origin of the wine.

Yeast: A multiclass classification dataset for predicting the cellular localisation sites of proteins.

Magic Gamma: A binary classification dataset used to discriminate between gamma-ray leaks into the atmosphere, and natural background radiation from a telescopic camera.

3.2.2 Data processing

Before we use the data in our experiments, we first normalise the data so that each feature lies on a common scale. We use standard normalisation on each feature because it is one of the most widely used methods of normalisation for machine learning problems and Siddiqui et al. (2015, 2019) used it to normalise their data before they computed their SFEs. For a data point $\mathbf{x}$, each feature $\mathbf{x}_i$ undergoes the following transformation:

$$\mathbf{x}_i = \frac{\mathbf{x}_i - \mu_i}{\sigma_i}, \tag{3.1}$$

where μ_i and σ_i represent the mean and standard deviation of the i^{th} feature respectively.

3.3 Notation

In the remainder of the thesis, the following notation is used, unless we specify otherwise:

- Let $\mathbf{x} \in \mathbf{X}$ represents a d-dimensional data point from the anomaly benchmark dataset $\mathbf{X}$ that contains n data points. We present the i^{th} data point in $\mathbf{X}$ by $\mathbf{x}^{(i)}$.
- Let $\mathcal{S}$ be any set of the feature indices. Then we denote $\mathbf{x}_\mathcal{S}$ as the projection of the data point $\mathbf{x}$ onto the feature subspace specified by $\mathcal{S}$, while $\mathbf{x}$ represents the data point in the full feature space.
- $SE(\mathbf{x})^k = (e_1, e_2, \ldots, e_k)$, represents an ordered SE list of size $k < d$ for the data point $\mathbf{x}$, where each e_i represents a subset of feature indices of size i. e_1 contains the first best feature index to be presented to the analyst, e_2 contains the two best feature indices to present to the analyst and so on.
- $SE(\mathbf{x}) = SE(\mathbf{x})^d$, is the full SE for $\mathbf{x}$.
- $SE(\mathbf{x})_i = e_i$, is the i^{th} explanation of size i in $SE(\mathbf{x})$.
- $SE(\mathbf{x})_0 = \emptyset$.

3.4 Outlier-based Sequential Explanations

Outlier-based methods compute SEs by using an outlier scoring measure guided by a search algorithm to identify the features that make the data point of interest outlying. An outlier scoring measure $\phi : \mathbb{R}^{|\mathcal{S}|} \longrightarrow \mathbb{R}$ is a measure of outlyingness that assigns a real

number to the data point $\mathbf{x}$ in feature subspace S. In this section, we outline the outlier-based SE methods. Section 3.4.1 presents the different anomaly detector outlier scoring measures used to measure the outlyingness of the features, while Section 3.4.2 presents the different search methods we will use to guide the computation of the outlier-based SEs.

3.4.1 Anomaly detector outlier scoring measures

There has been a variety of unsupervised anomaly detection algorithms developed over the years (Chandola et al. 2009). We selected a few unsupervised anomaly detection algorithms in our work that cover different approaches to scoring outliers. The selection of algorithms is not exhaustive, but it is a good representation of the field by trying to cover several different solution approaches that cover different assumptions and properties when scoring outliers. We chose the statistical, model based, nearest-neighbour, clustering and isolation based approaches, which cover the general methods in anomaly detection (Emmott et al. 2015). We use the outlier scoring measure of the anomaly detection algorithms to compute the outlier-based SEs. This section summaries each of the anomaly detection algorithms and how they score the outlyingness of a data point.

3.4.1.1 Kernel density estimation

A Kernel density estimation (KDE) (Silverman 1986) is a non-parametric statistical method used to estimate the probability density function (pdf) of a dataset. KDEs do this by using a kernel function to estimate the local densities of each data point in the dataset and then obtain the overall pdf by aggregating the local data densities of the data points. The outlier score of a data point is the probability of the data point not belonging to the fitted pdf. More formally, the KDE outlier scoring measure of a data point $\mathbf{x}$ in the feature subspace $\mathcal{S}$ is defined as (Silverman 1986) :

$$\phi(\mathbf{x}_{\mathcal{S}}) = 1 - \frac{1}{n(2\pi)^{|S|/2}\prod_{s\in\mathcal{S}} h_s} \sum_{\mathbf{z}\in\mathbf{X}_{\mathcal{S}}\setminus\{\mathbf{x}_{\mathcal{S}}\}} e^{-\sum_{s\in\mathcal{S}} \frac{(\mathbf{x}_s-\mathbf{z})^2}{2h_s^2}}, \tag{3.2}$$

where h_s is the bandwidth of feature $s \in \mathcal{S}$.

3.4.1.2 Ensemble Gaussian Mixture model

The Ensemble Gaussian Mixture Model (EGMM) (Glodek et al. 2013) is a statistical method that uses an ensemble of Gaussian Mixture Models (GMM) Glodek et al. (2013)

in order to improve the precision and stability of the estimated pdf rather than using a single GMM (Glodek et al. 2013). EGMMs have shown promising results for classifications and the approximation of non-Gaussian PDFs (Glodek et al. 2013). A GMM is a linear combination of K Gaussians (Bishop 2007), where the probability of a data point $\mathbf{x}$ in subspace S belonging to the distribution is:

$$P(\mathbf{x}_S) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_S | \mu_k, \Sigma_k), \tag{3.3}$$

where $\pi_k \in [0, 1]$ is the weight of the k^{th} Gaussian with parameters mean μ_k and covariance Σ_k and $\sum_{k=1}^{K} \pi_k = 1$. The parameters of μ_k, Σ_k and π_k are estimated by using the expectation-maximisation algorithm (Bishop 2007).

An EGMM is an ensemble of GMMs, and it is created as follows (Glodek et al. 2013):

1. Generate L GMMs represented by $g_1, \ldots, g_L$. To compute the L GMMs, the following steps are followed for each GMM g_i:
   (a) The training set is constructed by randomly sampling 70% of the original dataset $\mathbf{X}$;
   (b) Determine the number of mixture components K_i, where K_i is the number of mixture components of GMM g_i;
   (c) For each mixture component k randomly initialise μ_k,Σ_k and π_k using the k-means algorithm.
2. Compute all GMMs $g_1, \ldots, g_L$ by using the expectation-maximisation algorithm;
3. Compute the log-likelihoods $l_1, \ldots, l_L$ of $g_1, \ldots, g_L$ on the dataset $\mathbf{X}$;
4. Select the top $M \leq L$ GMMs with the highest log-likelihoods;
5. Compute the weights of each GMM $w_1, \ldots, w_M$ by using $w_i = \frac{\exp(l_i)}{\sum_{j=1}^{M} \exp(l_j)}, for\ i = 1, \ldots, M$;
6. The final EGMM is $g = \sum_{i=1}^{M} w_i g_i$.

To obtain the outlier score of a data point $\mathbf{x}$ in feature subspace $\mathcal{S}$, the EGMM is trained on $\mathbf{X}$ to obtain g and the marginal density of the EGMM on feature subspace $\mathcal{S}$ is represented by $g_{\mathcal{S}}$. The outlier score is, therefore, the probability that the data point $\mathbf{x}$ in feature subspace $\mathcal{S}$ does not belong to the fitted distribution and is defined as follows:

$$\phi(\mathbf{x}_{\mathcal{S}}) = 1 - g_S(\mathbf{x}_{\mathcal{S}}) \tag{3.4}$$

3.4.1.3 One-Class Support Vector Machine

The One-Class Support Vector Machine (OCSVM) (Schölkopf et al. 2001) is a model based anomaly detection method. Similar to an SVM for classification, an OCSVM transforms a dataset in the feature space $\mathbb{R}^d$ to a higher-dimensional feature space $\mathbb{R}^p$, where $d < p$. In the transformed space, the OCSVM maximises the distance from the hyperplane to the origin, which results in learnt regions that contain the training dataset in the original space. For anomaly detection, we assume that a vast majority of the data is normal and that anomalies contribute less to the computation of the region where the training data lies. The outlier score of each data point is the distance of the data point to the learnt region. Data points that fall within the region are considered to be inliers and have negative scores, whereas, data points outside the region have positive scores. More formally, the OSVM problem is reduced to the following optimisation problem:

$$\min_{\mathbf{w},\xi_i,\rho} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho; \tag{3.5}$$

$$\text{subject to:} \tag{3.6}$$

$$\mathbf{w}\cdot\theta(\mathbf{x}_{\mathcal{S}}^{(i)}) \geq \rho - \xi_i \text{ for all } i = 1,\ldots,n; \tag{3.7}$$

$$\xi_i \geq 0 \text{ for all } i = 1,\ldots,n, \tag{3.8}$$

where $\mathbf{w}$ is the normed vector of the hyperplane, θ is a kernel map that transforms the training examples into the space $\mathbb{R}^p$, ξ_i are nonzero slack variables, ν is an upper bound on the fraction of outliers and a lower bound of the fraction of support vectors, and ρ is the intercept of the hyperplane.

Using Lagrange techniques and using a kernel function K for the dot-product calculations, the OCSVM outlier scoring function of a data point $\mathbf{x}$ in the feature subspace $\mathcal{S}$ is:

$$\phi(\mathbf{x}_{\mathcal{S}}) = (\mathbf{w}\cdot\theta(\mathbf{x}_{\mathcal{S}}^{(i)}) - \rho) = \sum_{i=1}^{n}\alpha_i K(\mathbf{x}_{\mathcal{S}}, \mathbf{x}_{\mathcal{S}}^{(i)}) - \rho \tag{3.9}$$

, where $\mathbf{x}_{\mathcal{S}}^{(i)} \in X_{\mathcal{S}}$, α_i are the Lagrange multipliers and $K(\mathbf{x}_{\mathcal{S}}, \mathbf{x}_{\mathcal{S}}^{(i)}) = \theta(\mathbf{x}_{\mathcal{S}})^T\theta(\mathbf{x}_{\mathcal{S}}^{(i)}), \forall i = 1, 2, \ldots, n$.

3.4.1.4 Local outlier factor

Local outlier factor (LOF) (Breunig et al. 2000) is a nearest neighbourhood based anomaly detect method. For a data point $\mathbf{x}$, the LOF algorithm compares its local density to the local densities of its k-nearest neighbours (k-nn). A data point will get a high outlier score if its density is significantly lower than the density of its k-nn (Breunig et al. 2000). More formally the LOF outlier scoring function of a data point $\mathbf{x}$ in the feature subspace S is formulated as follows:

Let $\Upsilon(\mathbf{x}_\mathcal{S})$ be the distance of the data point $\mathbf{x}$ to its k^{th} nearest neighbour in the feature subspace specified by $\mathcal{S}$. Then we define the reachability distance between $\mathbf{x}_\mathcal{S}, \mathbf{t}_\mathcal{S} \in X_\mathcal{S}$ as:

$$\Omega_k(\mathbf{x}_\mathcal{S}, \mathbf{t}_\mathcal{S}) = max\{\Upsilon(\mathbf{t}_\mathcal{S}), d(\mathbf{x}_\mathcal{S}, \mathbf{t}_\mathcal{S}))\} \tag{3.10}$$

and the local reachability density of $\mathbf{x}_\mathcal{S}$ is defined by

$$\Lambda(\mathbf{x}_\mathcal{S}) = 1/(\frac{\sum_{\mathbf{t} \in k\text{-nn}(\mathbf{x}_\mathcal{S})} \Omega_k(\mathbf{x}_\mathcal{S}, \mathbf{t})}{\mid k\text{-nn}(\mathbf{x}_\mathcal{S}) \mid}). \tag{3.11}$$

Then the LOF outlier score of $\mathbf{x}_\mathcal{S}$ is:

$$\phi(\mathbf{x}_\mathcal{S}) = \frac{\sum_{\mathbf{t} \in k\text{-nn}(\mathbf{x}_\mathcal{S})} \frac{\Lambda(\mathbf{t})}{\Lambda(\mathbf{x}_\mathcal{S})}}{\mid k\text{-nn}(\mathbf{x}_\mathcal{S}) \mid} \tag{3.12}$$

3.4.1.5 k-means anomaly detector

The k-means anomaly detector is a clustering based anomaly detector that works on the assumptions that normal data points lie close to their closest cluster centroid, while outliers are far away from it (Chandola et al. 2009). The anomaly detector first clusters the data into k clusters using Lloyds algorithm (Kanungo et al. 2002), then it assigns each data point the distance from its closest cluster as its outlier score (Chandola et al. 2009). Data points with the largest distances from their closest cluster centroids are considered to be outliers because they do not belong to a specific cluster. More formally, the k-means outlier scoring measure of a data point $\mathbf{x}$ in the feature subspace $\mathcal{S}$ is formulated as follows:

$$\phi(\mathbf{x}_\mathcal{S}) = \|\mathbf{x}_\mathcal{S} - c_{\mathbf{x}_\mathcal{S}}\|_2, \tag{3.13}$$

where $c_{\mathbf{x}_\mathcal{S}}$ represents the data point $\mathbf{x}_\mathcal{S}$'s closest cluster centroid and $\|\cdot\|$ is the Euclidean distance or L_2 vector norm.

3.4.1.6 Isolation Forests

Isolation Forests (iTrees) (Liu et al. 2012) isolates data point by first randomly selecting a feature and a random split value between the maximum and minimum values of the selected feature in order to partition the data point into the region it belongs in. The process of randomly partitioning the data point continues until it is entirely isolated. Since a tree structure can represent recursive partitioning, the number of splittings required to isolate a data point is equivalent to the path length from the root node to the terminating node. Averaging the path length over a forest of random trees forms a measure of normality. Random partitioning produces shorter paths for outliers. Therefore, when a forest of random trees collectively produces shorter path lengths for particular data points, they are highly likely to be outliers. More formally, the iTree outlier scoring measure of a data point $\mathbf{x}$ in the feature subspace $\mathcal{S}$ is formulated as follows:

$$\phi(\mathbf{x}_\mathcal{S}) = -2^{\frac{E(h(\mathbf{x}_\mathcal{S}))}{c(\psi)}}, \tag{3.14}$$

where $h(\mathbf{x}_\mathcal{S})$ is the path length of $\mathbf{x}_\mathcal{S}$, $c(\psi)$ is the average path length of an unsuccessful search in a binary search tree, and ψ is the subsample size.

3.4.2 Sequential search outlier-based SEs

3.4.2.1 Sequential forward selection

Forward selection: Forward selection (FS) is a greedy sequential search method that adds one feature at a time to the previously selected subset of features in the SE. At each step, it adds the feature that maximises the outlier score of the previously selected features in the SE. More formally, FS computes the following outlier-based SE:

$$SE(\mathbf{x})_i = SE(\mathbf{x})_{i-1} \cup \underset{j \in \overline{SE}(\mathbf{x})_{i-1}}{argmax} \; \phi(\mathbf{x}_{SE(\mathbf{x})_{i-1} \cup j}), \tag{3.15}$$

where $\overline{SE}(\mathbf{x})_i$ are the subset of features not selected in $SE(\mathbf{x})_i$.

Independent forward selection: Independent forward selection (IND-FS) requires the computation of the outlier score of each individual feature $\phi(\mathbf{x}_j)$ for $j = 1, \ldots, d$. To compute the SE, IND-FS sorts the features in increasing order of their scores $\phi(\mathbf{x}_j)$. More formally, IND-FS computes the following outlier-based SE:

$$SE(\mathbf{x})_i = SE(\mathbf{x})_{i-1} \cup \underset{j \in \overline{SE}(\mathbf{x})_{i-1}}{argmax} \ \phi(\mathbf{x}_j) \tag{3.16}$$

3.4.2.2 Sequential backward selection

Backward selection: Backward selection (BS) starts with a full set of features, and at each iteration removes the feature that returns the lowest decrease in the outlier score and adds it to the SE. More formally, BS computes the following explanation:

$$SE(\mathbf{x})_i = SE(\mathbf{x})_{i-1} \cup \underset{j \in \overline{SE}(\mathbf{x})_{i-1}}{argmin} \ \phi(\mathbf{x}_{\overline{SE}(\mathbf{x})_{i-1} \setminus j}) \tag{3.17}$$

Independent backward selection: For each feature $\mathbf{x}_i$ of the data point $\mathbf{x}$, independent backward selection (IND-BS) assigns it with a score of $\phi(\mathbf{x} - \mathbf{x}_i) - \phi(\mathbf{x})$, where $\mathbf{x} - \mathbf{x}_i$ denotes the removal of feature $\mathbf{x}_i$ from $\mathbf{x}$. The removal of the features with the lower scores make $\mathbf{x}$ appear most outlying. IND-BS sorts the features in increasing order of their scores to compute the explanation. More formally, IND-BS computes the following outlier-based SE:

$$SE(\mathbf{x})_i = SE(\mathbf{x})_{i-1} \cup \underset{j \in \overline{SE}(\mathbf{x})_{i-1}}{argmin} \ (\phi(\mathbf{x} - \mathbf{x}_j) - \phi(\mathbf{x})). \tag{3.18}$$

3.4.3 Meta-heuristic search outlier-based SE

3.4.3.1 Particle swarm optimisation

The sequential search algorithms compute SEs that suffer from the nesting problem. To combat the nesting effect, we introduce a novel Particle swarm optimisation (PSO) meta-heuristic search that restricts the search of the feature subspace to only subsets of a specific size instead of searching over a range of subset sizes. Therefore, when we require $SE(\mathbf{x})_i$, the PSO meta-heuristic will restrict the search to only feature subspaces of size i. The PSO meta-heuristic search is known for its fast convergence and its tendency to fall into sub-optimal solutions. In order to take the PSO out of sub-optimal solutions and

improve exploration, we include a mutation factor which also reduces the convergence rate.

Let $f : \mathbb{R}^{|\mathcal{S}|} \longrightarrow \mathbb{R}$ represent the fitness function to be optimised by the PSO search algorithm. Let P represent the total number of particles in the swarm. Let $pbest_j$ be the best-known position of particle j and let $gbest$ be the best-known position of the entire swarm. Let G represent the total number of generations. The PSO meta-heuristic algorithm to compute the best feature subspace of a fixed size of i is as follows:

Algorithm 1: Particle Swarm Optimisation algorithm for a fixed size subset of size i

Input: $\mathbf{x}, \mathbf{X}, i$

Output: $gbest$

for $j = 1$ **to** P **do**
- Initialise the particle p_j by sampling without replacement i features from the full set of features $\{1, \ldots, d\}$
- Initialise the particle's best position to its initial position: $pbest_j = p_j$
- **if** $f(\mathbf{x}_{p_j}) < f(\mathbf{x}_{gbest})$ **then**
  - Update the swarm's best known position: $gbest = p_j$

for g=*1* **to** G **do**
- **for** $j = 1$ **to** P **do**
  - Update(p_j)
  - **if** $f(\mathbf{x}_{p_j}) > f(\mathbf{x}_{pbest_j})$ **then**
    - Update the particle j's best known position: $pbest_j = p_j$
  - **if** $f(\mathbf{x}_{p_j}) > f(\mathbf{x}_{gbest})$ **then**
    - Update the swarm's best known position: $gbest = p_j$

The update of the particles is the most important part of the PSO algorithm. We update each particle p_j as follows:

Update(p_j):

1. Pool all the unique features from $p_j, pbest_j$ and $gbest$;
2. Randomly select i features from the unique pool of features;
3. Each of the i randomly selected features has a fixed probability of mutation m ;
4. Elements selected for mutation are assigned new feature values at random from all the features not included in the pool of unique features from step 1.

For outlier-based SEs, to compute $SE(\mathbf{x})_i$ of a data point $\mathbf{x}$ based on the PSO meta-heuristic we replace the fitness function f with the outlier scoring measure ϕ. After the PSO has undergone G generations, we set $SE(\mathbf{x})_i = gbest$.

3.4.4 Hyperparameter Optimisation

We now discuss how we chose the parameters of the outlier-based SEs. For each anomaly detection benchmark dataset we performed hyperparameter optimisation using grid search to find the parameters that return the highest area under the receiver operating characteristic curves (AUC ROC) (Hanley & McNeil 1982) in the full feature space. The optimal parameter in the full feature space is used to compute the outlier-based SEs.

For the k-means anomaly detector we use grid search to find the optimal k for $k = 2, \ldots, 8$. For each k, we calculated the AUC ROC using the ground truth of the benchmark. For each benchmark, we chose the k value that gave us the highest AUC ROC and used it to compute the SEs for the k-means outlier scoring measure. For the LOF anomaly detector, we grid searched on $k = 15, \ldots, 20$ for each anomaly detection benchmark dataset. For the bandwidth of the KDE we applied grid search on $[0.001, 0.01, 0.1, 1, 10]$ and also on Silvermans (Silverman 1986) and Scotts(Scott 2010) rule of thumbs. For OCSVM we used a linear kernel and a RBF kernel and applied grid search on $C = [0.01, 0.1, 1, 10]$ on both kernels and $\gamma = [0.001, 0.01, 0.1, 1]$ on the RBF kernel. For the EGMM anomaly detector, we used the same parameters as in (Siddiqui et al. 2015, 2019), while for iTrees, we used the default parameters recommended by Liu et al. (2012) of 100 trees and a sub-sampling size of 256.

For the PSO, we limited the total number of computations in each search space to 3000 by setting $G = 75$ and $P = 40$. We set the mutation factor for each feature to $m = 0.1$. Also, to decrease the number of computations, we cached and saved all of our results in a hash table stored in RAM so that if the PSO algorithm generates the same particles we do not need to compute the value of their fitness function again, instead we do a quick look up the in the hash table.

3.5 Sample-based Sequential Explanations

In this section, we present the sample-based SEs. Section 3.5.1 presents the sampling method which turns the problem of computing SEs into a feature section problem, while Section 3.5.2 presents the different feature selection methods we use to compute the sample-based SEs.

3.5.1 Sampling method

We first define the definitions of the k-distance and reference set.

k-distance The k-distance of a data point $\mathbf{x} \in \mathbf{X}$, denoted by k-distance($\mathbf{x}$), is the distance $d(\mathbf{x}, \mathbf{x}')$ between $\mathbf{x}$ and its k-th nearest neighbour $\mathbf{x}'$ in the data set $\mathbf{X}$.

reference set The reference set of a data point $\mathbf{x}$, denoted by $R_k(\mathbf{x})$, is the set of points $\mathbf{p} \in \mathbf{X}$ whose distance from $\mathbf{x}$ is less than or equal to k-distance($\mathbf{x}$). More formally the reference set is defined as follows:

$$R_k(\mathbf{x}) = \{\mathbf{p} \in \{\mathbf{X} \setminus \{\mathbf{x}\}\} | d(\mathbf{x}, \mathbf{p}) \leq k\text{-distance}(\mathbf{x})\}$$

For a data point $\mathbf{x}$, the following steps are followed to compute the inlier and outlier class of $\mathbf{x}$ (Micenková et al. 2013):

1. We define the sampled inlier class of $\mathbf{x}$ from $\mathbf{X} \setminus \{\mathbf{x}\}$ as a union of its reference set $R_k(\mathbf{x})$ and a set of randomly drawn data points from $\mathbf{X} \setminus \{\{\mathbf{x}\} \cup R_k(\mathbf{x})\}$ as:

$$\mathcal{I}(\mathbf{x}) = R_k(\mathbf{x}) \cup \{\mathbf{q}_j\}_1^r, \tag{3.19}$$

   where each $\mathbf{q}_j$ is randomly chosen from $\mathbf{X} \setminus \{\{\mathbf{x}\} \cup R_k(\mathbf{x})\}$ and $r = |R_k(\mathbf{x})|$.

2. We define the outlier class of $\mathbf{x}$ as:

$$\mathcal{O}(\mathbf{x}) = \{\mathbf{x}\} \cup \{\mathbf{z}_j\}_1^s, \tag{3.20}$$

   where each $\mathbf{z}_j$ is randomly sampled from $\mathcal{N}(\mathbf{x}, \lambda^2 I)$, $s = |\mathcal{I}(\mathbf{x})|$, I is the $d \times d$ identity matrix and $\lambda = \alpha \cdot \frac{1}{\sqrt{d}} \cdot k$-distance($\mathbf{x}$), and α is a user defined parameter which controls the width of the distribution.

3. We obtain $SE(\mathbf{x})_i$ by selecting the best i features that separate the inlier and outlier classes using a feature selection algorithm.

3.5.2 Feature selection methods

3.5.2.1 Wrapper-based methods

Forward selection SVM: Forward selection SVM (FS-SVM) (Chandrashekar & Sahin 2014) makes use of the SFS sequential search method to greedily add features into its SE while being guided by the classification accuracy of the SVM classifier. At each iteration it uses 3-fold cross-validation to train the classifier on the inlier and outlier classes of $\mathbf{x}$ and adds the feature in the SE that returns the highest average classification

accuracy over the 3 folds. In order to obtain $SE(\mathbf{x})_i$, the algorithm starts with an empty set. At each iteration, it classifies the training data over the 3 cross-validation folds using an SVM and adds the feature that maximises the average classification accuracy over the 3 folds with the current subset of already chosen features into $SE(\mathbf{x})_i$ until i features are selected.

Backward selection SVM: Backward selection SVM (BS-SVM) (Chandrashekar & Sahin 2014) makes use of the SBS sequential search method to add features into its SE greedily. In order to obtain $SE(\mathbf{x})_i$, the algorithm starts with the full set of features. At each iteration, it removes the feature that returns the lowest decrease in the average classification accuracy of the SVM over 3 cross-validation folds and adds it into $SE(\mathbf{x})_i$ until i features are selected.

PSO-SVM: The PSO-SVM uses the PSO meta-heuristic from Algorithm 1 to search for SEs that are not nested using the accuracy of the SVM model as the fitness function f. The fitness function f of the PSO-SVM is simply the average classification accuracy of the SVM that separates the inlier and outlier classes over 3 cross-validation folds.

3.5.2.2 Embedded-based methods

SVM recursive feature elimination: SVM recursive feature elimination (SVM-RFE) (Guyon et al. 2002) learns which features are influential to the accuracy of the SVM model. For a linear kernel SVM-RFE, each feature i is given the following score: $J(i) = \mathbf{w}_i^2$, where $\mathbf{w}_i$ is the i^{th} element of $\mathbf{w}$ which is the normal vector to the hyperplane of the SVM (Guyon et al. 2002). While for non-linear kernel functions the score is given by $J(i) = \frac{1}{2}\alpha^T\mathbf{H}\alpha - \frac{1}{2}\alpha^T\mathbf{H}(-i)\alpha$, where $\mathbf{H}$ is a matrix with elements $y^{(h)}y^{(k)}K(\mathbf{x}^{(h)}, \mathbf{x}^{(h)})$, $H(-i)$ represents the matrix $\mathbf{H}$ with the feature i removed and α represents the Lagrange multipliers. At each iteration, we train an SVM model over 3 cross-validation folds and remove the feature with the smallest average score because it has the least effect on the classification accuracy of the SVM (Chandrashekar & Sahin 2014). The next iteration only uses the remaining features. The process continues until there are no more features remaining (Guyon et al. 2002). The final step of the algorithm is to sort the features in the reverse order of their removal and $SE(\mathbf{x})_i$ is obtained by selecting the first i features in the sorted list.

Random forests recursive feature elimination: RFs estimates the importance for each feature in every tree by permuting the data for that feature and measuring how

much the OOB error increases while all the other features are left unchanged (Liaw & Wiener 2002). The importance score of a feature is the average the difference of the OOB error before and after the permutation over all the trees in the RF (Liaw & Wiener 2002). At each iteration, Random forests recursive feature elimination (RF-RFE) (Liaw & Wiener 2002) removes the feature that has the lowest average score over 3 cross-validation folds, while the remaining features remain for the next iteration. The process continues until there are no more features remaining. The final step of the algorithm is to sort the features in the reverse order of their removal and $SE(\mathbf{x})_i$ is obtained by selecting the first i features in the sorted list.

3.5.2.3 Filter-based methods

Maximum relevance minimum redundancy: Maximum relevance minimum redundancy (mRmR) (Radovic et al. 2017, Peng et al. 2005) is an algorithm developed for feature selection of micro-array data and selects a subset of features having the most correlation with a class (relevance) and the least correlation between themselves (redundancy). It is an information theoretic method that uses mutual information (Brown et al. 2012) to measure the dependency between features. More formally, $SE(\mathbf{x}_i)$ is obtained as follows:

$$SE(\mathbf{x})_i = SE(\mathbf{x})_{i-1} \cup \underset{j \in \overline{SE}(\mathbf{x})_{i-1}}{argmax} \; (I(\mathcal{X}(\mathbf{x})_j; \mathcal{Y}(\mathbf{x})) - \frac{1}{i-1} \sum_{k \in SE(\mathbf{x})_{i-1}} I(\mathcal{X}(\mathbf{x})_j; \mathcal{X}(\mathbf{x})_k)), \tag{3.21}$$

where $I(X;Y) = \sum_{x \in X, y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$ is the discrete mutual information between variables X and Y, $\mathcal{X}(\mathbf{x}) = \mathcal{I}(\mathbf{x}) \cup \mathcal{O}(\mathbf{x})$ is the union of the inlier and outlier class of $\mathbf{x}$ and $\mathcal{Y}(\mathbf{x})$ is the label, where $\mathcal{X}(\mathbf{x})_j$ represents the jth feature vector in $\mathcal{X}(\mathbf{x})$.

Mutual information feature selection: Mutual information feature selection (MIFS) (Radovic et al. 2017, Peng et al. 2005) is an information theoretic feature selection method that focuses on reducing redundancy (Battiti 1994). More formally, $SE(\mathbf{x}_i)$ is obtained as follows:

$$SE(\mathbf{x})_i = SE(\mathbf{x})_{i-1} \cup \underset{j \in \overline{SE}(\mathbf{x})_{i-1}}{argmax} \; (I(\mathcal{X}(\mathbf{x})_j; \mathcal{Y}(\mathbf{x})) - \beta \sum_{k \in SE(\mathbf{x})_{i-1}} I(\mathcal{X}(\mathbf{x})_j; \mathcal{X}(\mathbf{x})_k), \tag{3.22}$$

where β is a configurable value.

3.5.3 Hyperparameter Optimisation

In this section, we discuss how we optimise the parameters that require optimisation in the sample-based SEs. To create the inlier and outlier class for a data point $\mathbf{x}$, we set its k for the reference set $R_k(\mathbf{x})$ to $k = 0.1 \times n$ and set $\alpha = 0.35$ as suggested by Micenková et al. (2013) . For each data points inlier and outlier class, we optimise the parameters of the feature selection methods that will compute its sample-based SEs by using the grid search method and selecting the parameters that return the highest average classification accuracy using 3-fold cross-validation in the full feature space.

For the feature selection methods that use SVM, we use the grid search method over the same parameters as in Section 3.4.4 and selected the parameters that return the highest average classification accuracy using 3-fold cross-validation in the full feature space to compute the SEs. For MIFS, we used $\beta = 1$ as suggested by Battiti (1994) and used 100 trees for RF-RFE, while mRmR did not require any parameters.

3.6 Performance Evaluations

The anomaly detection benchmark datasets do not come with ground truth explanations or an analyst who can evaluate the features generated by the SEs. Hence, we model a simulated analyst as a conditional distribution of the anomaly class given a subset of features (Siddiqui et al. 2015, 2019). More formally, we model the simulated analyst as the function $A(\mathbf{x}, \mathcal{S}) = P(\mathbf{x} = anomaly|\mathbf{x}_\mathcal{S})$. $A(\mathbf{x}, \mathcal{S})$ returns the probability of the data point $\mathbf{x}$ being anomalous considering only the features specified in the feature space $\mathcal{S}$.

Given $SE(\mathbf{x})^k$, we define the analyst certainty curve for the first $k < d$ feature subsets of the SE of $\mathbf{x}$ as the curve that plots the coordinates $(i, A(\mathbf{x}, SE(\mathbf{x})_i)$ for $i = 1, \ldots, k$. Intuitively, the higher the analyst certainty curve the better the quality of the SE because the selected feature subsets in the SE are returning high probabilities of the analyst detecting the anomalies given the features presented to them. To measure the performance of the SE, we measure the area under the curve of the analyst certainty curve (AUCC) and divide it by k so that the AUCC is between 0 and 1.

We now specify how we obtain the analyst function $A(\mathbf{x}, \mathcal{S})$. Since our anomaly detection benchmarks are labelled, we use supervised learning to construct the simulated analyst by constructing a training set consisting of the normal and anomaly classes. For the Emmott et al. (2015) datasets we construct the training set over the motherset, while for the Goldstein & Uchida (2016) datasets we constructed the training set over the actual anomaly benchmark datasets because we do not have access to the motherset datasets.

We used a brute force method to obtain $A(\mathbf{x}, \mathcal{S})$ by training a discriminative model only in the feature space specified by $\mathcal{S}$. It leads to multiple trained models because we train a model for every feature space $\mathcal{S}$ encountered in the SEs. For each model we train, we exclude that data point whose SEs we are evaluating in the training set so that we avoid any bias in our probability estimations. In order to obtain reliable "true" probability estimates even in the case of imbalanced datasets, we use a Calibrated Bagged Random Forest (CBRF) (Wallace & Dahabreh 2012, 2014) model. To obtain $A(\mathbf{x}, \mathcal{S})$, CBRFs work as follows:

1. Use bootstrap aggregating over the normal class in feature subspace $\mathcal{S}$ to obtain 30 new normal training datasets each the size of the anomalous class. Now we have 30 new balanced datasets each consisting of the new normal training set and the anomalous class in feature subspace $\mathcal{S}$.

2. Fit a RF consisting of 100 trees on each of the 30 bootstrap datasets. There are now 30 predictive RF models for the feature subspace $\mathcal{S}$. Let $f_j(\mathbf{x}_\mathcal{S})$, for $j \in 1, \ldots, 30$ represent the predictive value of model j for the data point $\mathbf{x}_\mathcal{S}$.

3. To calibrate each model and transform its outputs into true probabilities, Wallace & Dahabreh (2012, 2014) assume that for each models predictive value $f_j(\mathbf{x}_\mathcal{S})$, there exists $\hat{\beta}$ coefficients that transform it into true probability. The estimate of the probability for data point $\mathbf{x}_\mathcal{S}$ according to model j is then:

$$\hat{P}_j(y|f_j(\mathbf{x}_\mathcal{S})) = \frac{1}{1 + e^{-\hat{\beta}_{0j} - \hat{\beta}_{1j} f_j(\mathbf{x}_\mathcal{S})}}, \tag{3.23}$$

   where the $\hat{\beta}$ coefficients are estimated using a maximum likelihood method that optimises on the same training set as that for the original model j and y is the binary class label representing the normal or anomaly class.

4. The calibrated true probability aggregated from all the 30 models is then represented by

$$\hat{P}(y|\mathbf{x}_\mathcal{S}) = \frac{1}{z} \sum_{j=1}^{30} \frac{1}{\text{Var}(\hat{P}_j(y|f_j(\mathbf{x}_\mathcal{S})))} \hat{P}_j(y|f_j(\mathbf{x}_\mathcal{S})), \tag{3.24}$$

   where $z = \sum_{j=1}^{30} \frac{1}{\text{Var}(\hat{P}_j(y|f_j(\mathbf{x}_\mathcal{S})))}$ is a normalisation constant.

5. $A(\mathbf{x}, \mathcal{S})$ is therefore the probability of the data point $\mathbf{x}_\mathcal{S}$ belonging to the anomalous class in the trained CBRF:

$$A(\mathbf{x}, \mathcal{S}) = \hat{P}(y = \text{anomaly}|\mathbf{x}_\mathcal{S}) \tag{3.25}$$

$$= P(\mathbf{x} = \text{anomaly}|\mathbf{x}_\mathcal{S}). \tag{3.26}$$

Chapter 4

Results and Discussions

In this chapter, we present and discuss the results of our study. We ran our experiments on multiple servers consisting of $16, 32$ and 64 CPUs and also $16, 32$ and 64 gigabytes of RAM. Each SE method was run five times, and the results that we present throughout this chapter represent the average over the five runs.

4.1 Results guideline

4.1.1 AUCC tables

Tables 4.1 and 4.2 contain the average AUCCs results of Emmott et al. (2015) datasets for the outlier and sample-based SEs respectively. Whereas, Tables 4.3 and 4.4 contain the AUCCs results of the Goldstein & Uchida (2016) datasets for the outlier and sample-based SEs respectively. For the Goldstein & Uchida (2016) anomaly benchmark datasets, we averaged the AUCCs of the anomalies to obtain the benchmarks average AUCC. Whereas for the (Emmott et al. 2015) datasets we first averaged the AUCCs of the anomalies for each benchmark to obtain its average AUCC, after that, we averaged the average AUCCs of each benchmark dataset which was constructed from the same motherset to obtain an average of the AUCCs of the motherset's anomaly detection benchmark datasets.

4.1.2 Analyst certainty curves

Column (a) of Figures 4.4 and 4.5 and Figure 4.6 contains the analyst certainty curves of the top performing anomaly detector (AD) outlier scoring measures and search pairs for each AD outlier scoring measure for the Emmott et al. (2015) and Goldstein & Uchida

(2016) datasets. While Column (b) contains the analyst certainty curves of the sample-based SEs. Interested readers are encouraged to refer to Appendices A and B for the analyst curves of the rest of the outlier-based SE results presented in Tables 4.1 and 4.3 for the Emmott et al. (2015) and Goldstein & Uchida (2016) datasets respectively. Each analyst curve contains error bars which represent a 95% confidence interval of the results over the five runs per SE. For the Emmott et al. (2015) datasets we generated the SEs $SE(\mathbf{x})_i, \forall i = 1, \cdots, d$ until the analyst is presented with all the features. Doing this enables us to analyse the analyst's conditional probability from the first feature subset up until the full set of features is presented. Whereas, for the Goldstein & Uchida (2016) datasets, we generated the SEs up to the first seven feature subsets because the dimensionality of the anomaly benchmark datasets is large.

4.2 Random and Oracle SEs

In addition to the SEs presented in the methodology, we compute the Random and Oracle SEs. A Random SE is a SE that randomly presents features to the analyst. Random SEs provide a lower bound on the attainable performance of our SEs. An OptOracle SE is an optimal SE that is allowed access to the simulated analyst, where $SE(\mathbf{x})_i$ is the set of i features that maximise the simulated analyst's conditional probability $A(\mathbf{x}, SE(\mathbf{x})_i) = P(\mathbf{x} = \text{anomaly}|\mathbf{x}_{SE(\mathbf{x})_i})$. Lastly, an anomaly detector's Oracle (AD Oracle) SE is an outlier-based SE, where $SE(\mathbf{x})_i$ is the set of i features that maximise the anomaly detector's outlier scoring measure. An AD Oracle provides the best outlier-based SE that an AD's outlier scoring measure can compute.

4.3 Analysing the results using the analyst certainty curves and AUCCs

4.3.1 Analyst certainty curves versus AUCCs

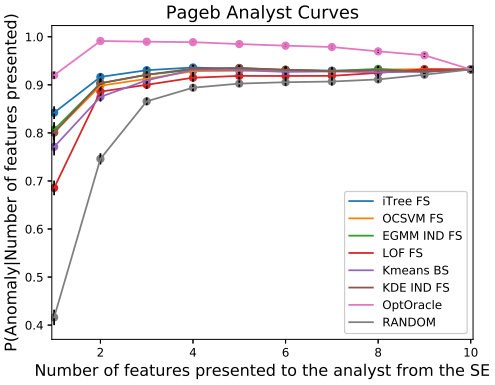


FIGURE 4.1: Top outlier-based Pageb analyst curves

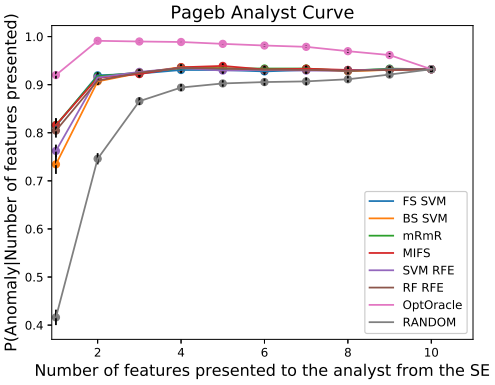


FIGURE 4.2: Sampled-based Pageb analyst curves

Figures 4.1 and 4.2 present the top performing outlier-based SEs and the sample-based SEs for the Pageb motherset anomaly benchmarks respectively. The average AUCC of the Random SE across the Pageb motherset anomaly benchmarks is 0.7753, while the best performing outlier based SE, iTree FS has an average AUCC of 0.8329. Numerically, the difference between the average AUCCs of the Random and iTree FS SE is not significantly different. However, the analyst certainty curves of the iTree FS and Random SEs provide more insight into the performances of the SEs rather than relying on the average AUCCs alone. The first feature that iTree FS SE selects returns the simulated analyst with a conditional probability of 0.85 of being certain that the data points presented are anomalous. While the Random SE returns a probability of 0.42 for the first randomly presented feature shown to the analyst. Given the first feature from the iTree FS SE, the analyst may wish to stop the investigation and conclude that the data points are anomalous if the feature returns a probability of that exceeds the analyst's threshold probability of detection. Whereas, for the Random SE the analyst will have to request for more features to be presented in order to make a final decision because a probability of 0.42 of being anomalous does not provide the analyst with enough confidence that the data points presented are anomalous. Considering the next two features in the iTree FS and Random SEs, the conditional probability of the analyst's detection increases to 0.91 for the iTree FS SE, while it increases to 0.73 for the Random SE. The analyst's conditional probability for the Random SE monotonically increases with respect to the number of features presented. While for the iTree FS SE, the conditional probability

monotonically increases from the first up until it reaches its peak at the third feature subset in the SE and then remains constant until the SE presents the analyst with the full set of features. The AUCC provides a measure to compare which SEs perform better than the other, while a visual inspection of the analyst certainty curves provides insights on how the SE performs for each feature subset.

4.3.2 Types of analyst certainty curves

From the Emmott et al. (2015) analyst certainty curves, we observed three different types of analyst certainty curves. Type 1 curves are monotonically increasing with respect to the number of features presented to the simulated analyst until the SE presents the analyst with the full set of features. Type 2 curves are monotonically increasing with respect to the number of features presented to the simulated analyst up until a peak is reached, after that it monotonically decreases until the SE presents the analyst with the full set of features. Type 3 curves are monotonically increasing with respect to the number of features presented up until they reached a plateau that is equal to the conditional probability of the presentation of the full set of features.

All of the Random SEs are type 1 analyst curves. It shows that the analyst's certainty is an increasing function of the number of features randomly presented. Type 2 certainty curves show that the SE can find at least one feature subset explanation that returns a conditional probability that is higher than the probability of presenting the analyst with the full set of features. An example of type 2 SE curves is the Yeast motherset k-means, OCSVM IND-FS SEs and the Magic Gamma motherset iTree, KDE, LOF FS SEs. The Magic Gamma KDE FS SE for example peaks at the second feature subset explanation that is presented to the analyst with a conditional probability of 0.88 and slowly dips down to the full set of features which has a conditional probability of 0.85. The second feature subset explanation which contains only two features provides the analyst with more information about why the data points are anomalous compared to presenting the analyst the full set of features. The feature subsets in the SE following the second feature subset are uninformative and reduce the analyst's certainty. The outlier and sample-based SEs of the Shuttle and Wine mothersets are all Type 3 certainty curves. The Wine LOF FS SE, for example, starts to plateau at the third feature subset explanation. All of the subsequent feature subsets are redundant and do not affect the analyst's conditional probability because the features that explain the data points have been identified earlier in the explanation. The feature subsets identified after the third feature subset in the SE are redundant because the conditional probability of the analyst remains stagnant and equal to the conditional probability of the full set of features.

4.4 MFP compared to AUCC

We have shown that analysing the results using the AUCC alone is not enough. The AUCC give us a numerical measure to compare the SEs against each other. However, to obtain more insight into the SEs performance, the analyst certainty curves need to be analysed. Siddiqui et al. (2015, 2019) used MFPs to evaluate their SFEs. The one problem we identified with MFP evaluations is that they can view two SEs as being equal even though one of the SEs may be far superior to the other because it provides higher probabilities throughout the explanations. A good example of this is presented in Figure 4.3.

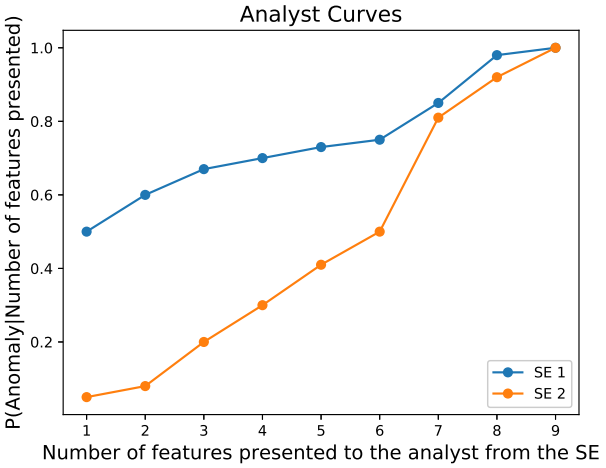


FIGURE 4.3: With a threshold of 0.8, SE 1 and 2 have an MFP of 7 and 8 if the threshold is 0.9. The AUCC of SE 1 is 0.67 and 0.46 for SE 2. The AUCC evaluation measure, along with analysing the analyst certainty curves captures the performance difference in the SEs much better than the MFP.

If the simulated analyst's threshold of detection is 0.8, then SE 1 and SE 2 would both have an MFP of 7. If the threshold is 0.9, then the MFP for both SEs would be 8. However, from observing the analyst certainty curves it clear that SE 1 identifies feature subsets that return higher conditional probabilities than SE 2. In addition, the AUCC of SE 1 is 0.67, whereas, the AUCC for SE 2 is 0.46. The AUCC captures the difference in performances of the SEs much better than the MFP. Evaluating the SEs using the AUCC along with the analyst certainty curve gives us more insight into the performance of the SEs for every feature subset. Looking at the Magic Gamma analyst certainty curves, we can observe that the OptOracle SE outperforms the rest of the SEs we generated by far, however, if we used MFPs to do the evaluation then the sampled-based FS-SVM SE and the OptOracle SE would both have an MFP of 1 if the threshold was 0.8. If we used MFPs to evaluate our results, we would get less insight from our results, and it

would be difficult to distinguish the performances of two or more SEs when they return the same MFPs.

TABLE 4.1: Outlier-based SE AUCCs of the Emmott et al. (2015) datasets. The bold black numbers represent the best performing search methods for each outlier scoring measure, while the bold red numbers represent the overall best performing methods per dataset.

Dataset	Anomaly Detector	IND FS	FS	IND BS	BS	AD Oracle	Opt Oracle	Random
Abalone	**iTree**	0.7257	0.7316	**0.7348**	0.7337	0.7502	0.8096	0.6966
	OCSVM	0.7138	0.7255	**0.7316**	0.7300	0.7447		
	EGMM	0.7212	**0.7298**	0.7133	0.7144	0.7448		
	LOF	0.7353	**0.7407**	0.7335	0.7296	0.7520		
	Kmeans	0.7223	0.7210	**0.7245**	0.7244	0.7450		
	KDE	0.7220	**0.7291**	0.7029	0.7025	0.7438		
Concrete	**iTree**	0.8051	**0.8156**	0.8044	0.8042	0.8290	0.8605	0.7231
	OCSVM	0.7480	0.7562	**0.7675**	0.7645	0.7800		
	EGMM	0.8006	**0.8113**	0.7948	0.8001	0.8280		
	LOF	0.7844	0.7908	0.7839	**0.7932**	0.7991		
	Kmeans	0.7642	**0.7938**	0.7800	0.7905	0.8053		
	KDE	0.7984	**0.8083**	0.7820	0.7875	0.8229		
Magic Gamma	**iTree**	0.7753	**0.7837**	0.7746	0.7733	-	0.8641	0.7250
	OCSVM	0.7513	**0.7616**	0.7595	0.7598			
	EGMM	0.7765	**0.7851**	0.7771	0.7723			
	LOF	0.7536	**0.7768**	0.7709	0.7645			
	Kmeans	0.7649	0.7685	**0.7714**	0.7688			
	KDE	0.7785	**0.7816**	0.7564	0.7630			
Pageb	**iTree**	0.8328	**0.8329**	0.8287	0.8300	-	0.8780	0.7753
	OCSVM	0.8163	**0.8266**	0.8162	0.8159			
	EGMM	**0.8287**	0.8269	0.8103	0.8110			
	LOF	0.8133	**0.8139**	0.8033	0.8025			
	Kmeans	0.8199	0.8194	0.8200	**0.8214**			
	KDE	**0.8285**	0.8242	0.8031	0.8045			
Shuttle	**iTree**	0.8574	0.8565	0.8614	**0.8615**	-	0.8834	0.7680
	OCSVM	0.8471	0.8064	0.8459	**0.8571**			
	EGMM	0.8661	**0.8670**	0.7269	0.7227			
	LOF	0.8477	**0.8673**	0.7983	0.7960			
	Kmeans	0.8677	**0.8694**	0.85106	0.8521			
	KDE	0.866	**0.8670**	0.615	0.6208			
Wine	**iTree**	0.7749	**0.7795**	0.7742	0.7752	-	0.8769	0.7305
	OCSVM	0.7525	0.757	**0.7664**	0.7635			
	EGMM	0.7628	**0.7794**	0.7618	0.7628			
	LOF	0.7754	**0.7830**	0.7638	0.762			
	Kmeans	0.7671	0.7717	**0.7729**	0.7716			
	KDE	0.7726	**0.7820**	0.7492	0.7555			
Yeast	**iTree**	0.6517	0.6168	0.6680	**0.6730**	0.6895	0.7992	0.5913
	OCSVM	0.6331	0.6483	**0.6515**	0.6293	0.6642		
	EGMM	0.6747	**0.6778**	0.6741	0.6736	0.6963		
	LOF	0.6410	**0.6526**	0.6309	0.6284	0.6701		
	Kmeans	**0.6687**	0.6655	0.6524	0.6578	0.6912		
	KDE	0.6787	**0.6851**	0.6706	0.6739	0.7035		

TABLE 4.2: Sample-based SE AUCCs of the Emmott et al. (2015) datasets. The bold red numbers represent the best performing feature selection method for each dataset.

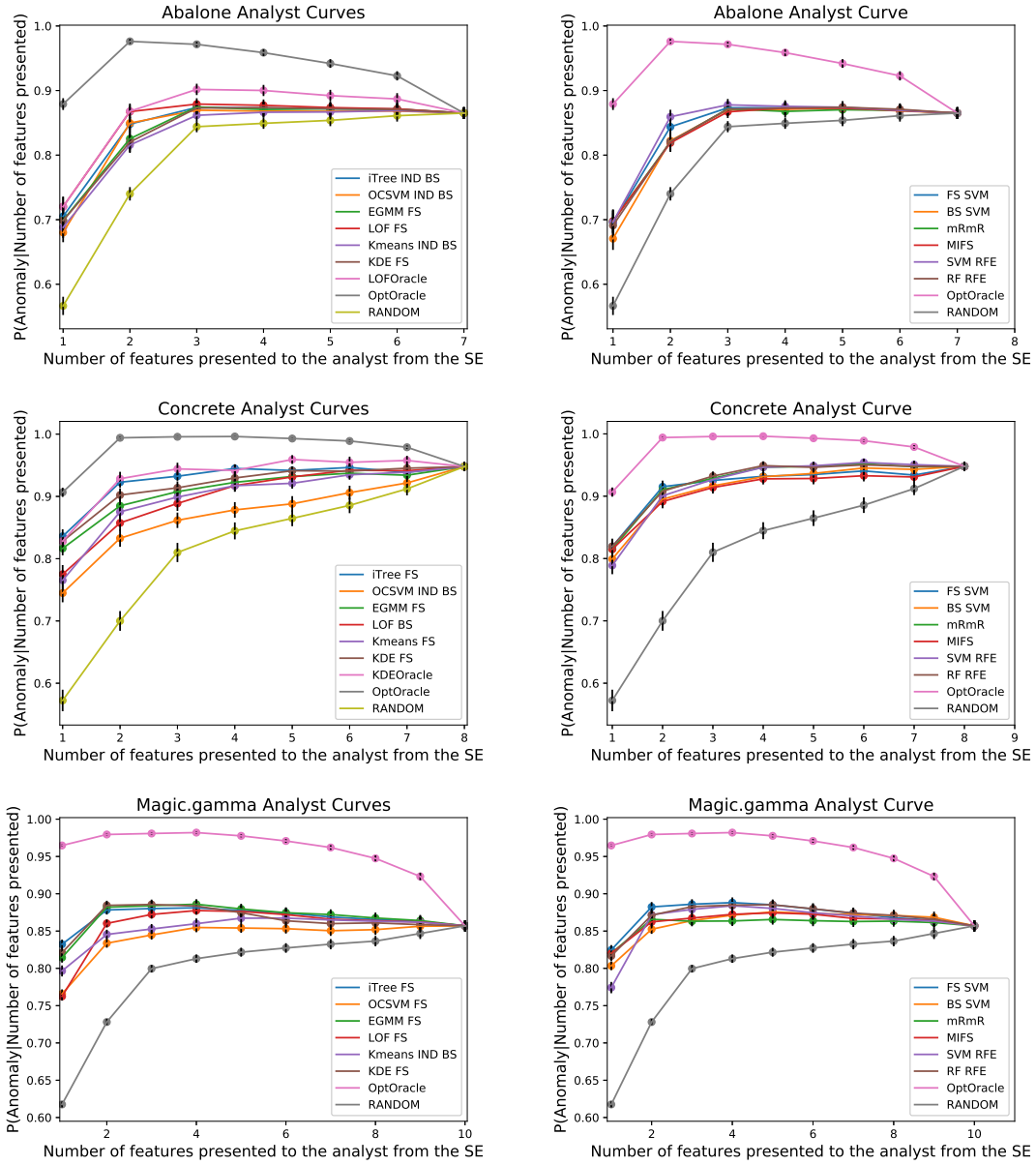
Dataset	FS SVM	BS SVM	mRmR	MIFS	SVM RFE	RF RFE	Opt Oracle	Random
Abalone	0.7334	0.727	0.7274	0.7279	**0.7372**	0.7289	0.8096	0.6966
Concrete	0.8089	0.8061	0.8152	0.8015	0.8132	**0.8156**	0.8605	0.7231
Magic Gamma	**0.7873**	0.7783	0.7752	0.7788	0.7813	0.7857	0.8641	0.7250
Pageb	0.8305	0.8271	**0.8322**	0.8319	0.8286	0.8300	0.8780	0.7753
Shuttle	0.8474	0.8397	0.8410	0.8545	**0.8667**	0.8616	0.8834	0.7680
Wine	0.7761	0.7659	0.7659	0.7748	0.7758	**0.7770**	0.8769	0.7305
Yeast	0.6629	0.6698	0.6266	0.6627	**0.6761**	0.6177	0.7992	0.5913

TABLE 4.3: Outlier-based SE AUCCs of the Goldstein & Uchida (2016) datasets. The bold black numbers represent the best performing search methods for each outlier scoring measure, while the bold red numbers represent the overall best performing methods per dataset.

Dataset	Anomaly Detector	IND FS	FS	IND BS	BS	AD Oracle	Random
Aloi	**iTree**	0.5269	**0.5705**	0.5617	0.5552	0.5810	0.4075
	OCSVM	0.5278	0.5381	**0.5574**	0.5528	0.5685	
	EGMM	0.5550	**0.5742**	0.5208	0.5343	0.5849	
	LOF	**0.5344**	0.5241	0.5273	0.5073	0.5486	
	Kmeans	0.5294	0.5552	**0.5589**	0.5524	0.5771	
	KDE	0.5597	**0.5602**	0.4626	0.4900	0.5858	
Annthyroid	**iTree**	**0.8429**	0.8385	0.8375	0.8423	0.8491	0.5659
	OCSVM	**0.7653**	0.7604	0.6890	0.6703	0.7906	
	EGMM	0.7808	**0.7896**	0.7374	0.7436	0.7995	
	LOF	**0.8392**	0.8296	0.7990	0.7935	0.8431	
	Kmeans	0.7795	0.7856	0.7599	**0.7911**	0.8088	
	KDE	0.7447	**0.7705**	0.6954	0.7116	0.7936	
Kdd99	**iTree**	0.8476	**0.8500**	0.8472	0.8482	0.8506	0.6913
	OCSVM	0.8465	**0.8493**	0.7550	0.7708	0.8500	
	EGMM	**0.8501**	0.8477	0.7139	0.7315	0.8527	
	LOF	0.8458	0.8398	**0.8475**	0.8461	0.8485	
	Kmeans	0.8385	**0.8411**	0.8215	0.8206	0.8428	
	KDE	**0.8459**	0.8449	0.7267	0.7520	0.8505	
Satellite	**iTree**	0.8226	**0.8250**	0.7827	0.7919	0.8307	0.6435
	OCSVM	0.7953	**0.8089**	0.7783	0.7770	0.8209	
	EGMM	0.8200	**0.8234**	0.7901	0.8001	0.8326	
	LOF	**0.7838**	0.7758	0.7562	0.7634	0.8049	
	Kmeans	0.8124	**0.8163**	0.7954	0.7735	0.8292	
	KDE	0.8031	**0.8118**	0.7481	0.7716	0.8275	

TABLE 4.4: Sample-based SE AUCCs of the Goldstein & Uchida (2016) datasets. The bold red numbers represent the best performing feature selection method for each dataset.

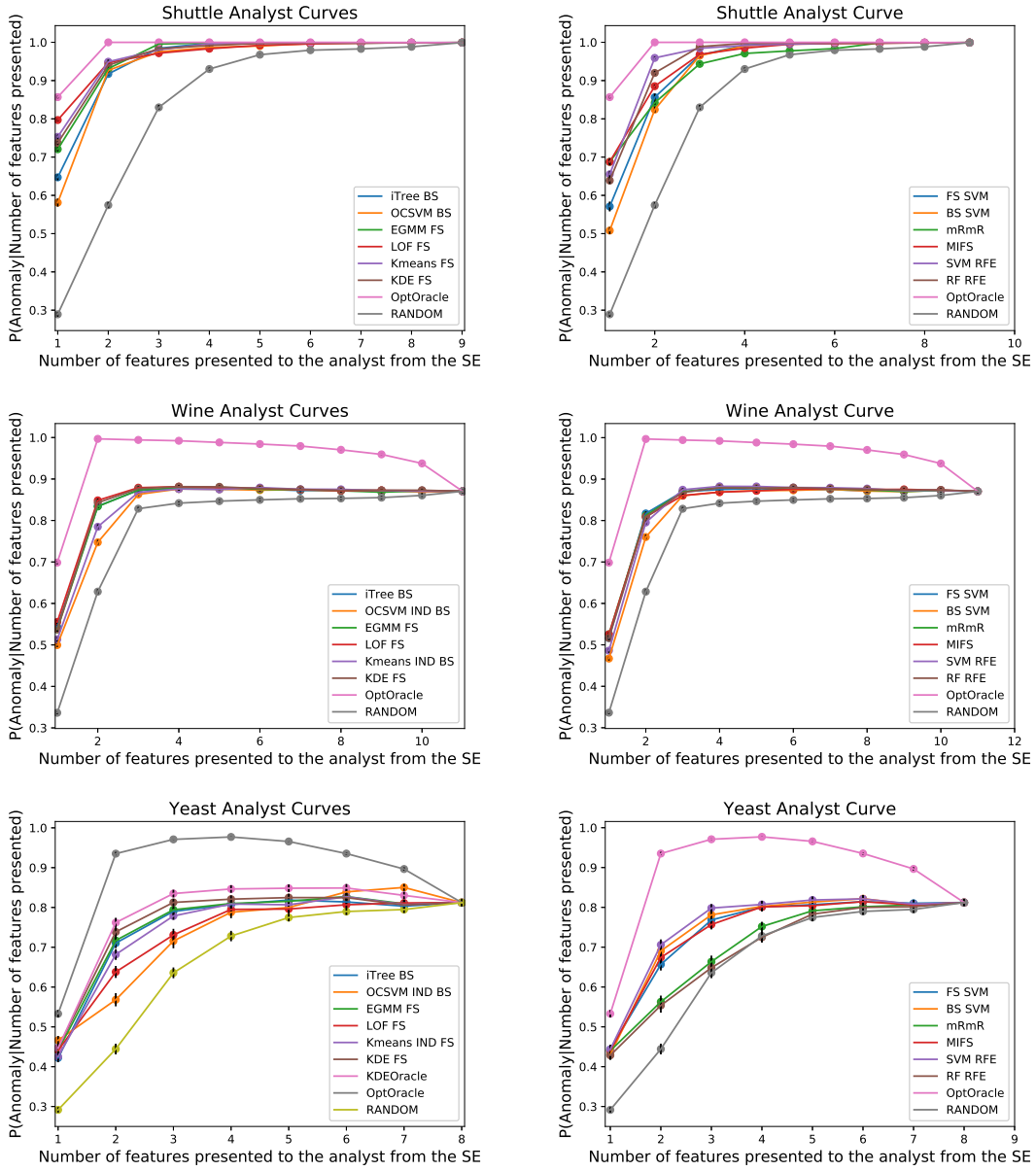
Dataset	FS SVM	BS SVM	mRmR	MIFS	SVM RFE	RF RFE	PSO SVM	Random
Annthyroid	0.8158	0.8130	0.8210	0.8217	**0.8348**	0.8143	0.8273	0.5659
Aloi	0.5656	0.5424	0.5065	0.5313	**0.5905**	0.5359	0.5742	0.4075
Kdd99	0.8069	0.7857	**0.8502**	0.8476	0.8490	0.8484	0.8145	0.6913
Satellite	0.8343	0.8343	0.8130	0.8069	0.8237	0.8218	**0.8368**	0.6435



Column (a): Top Outlier-based SEs

Column (b): Sample-based SEs

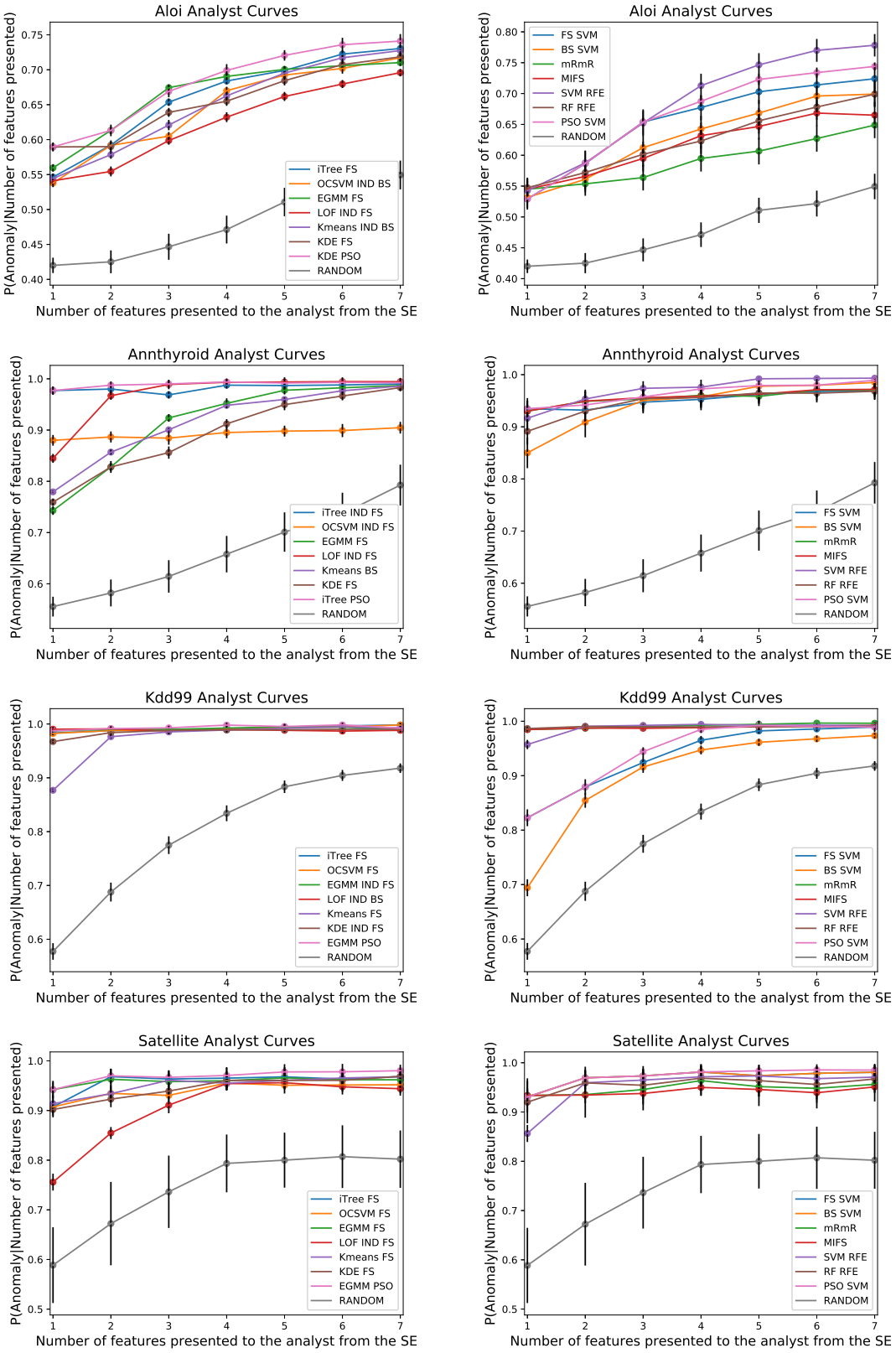
FIGURE 4.4: Top outlier-based SE analyst curves in Column (a) and sampled-base SE analyst curves in Column (b) for the Emmott et al. (2015) datasets



Column (a): Top Outlier-based SEs

Column (b): Sample-based SEs

FIGURE 4.5: Top outlier-based SE analyst curves in Column (a) and sample-based SE analyst curves in Column (b) for the Emmott et al. (2015) datasets



Column (a): Top Outlier-based SEs

Column (b): Sample-based SEs

FIGURE 4.6: Top outlier-based SE analyst curves in Column (a) and sample-based SE analyst curves in Column (b) for the Goldstein & Uchida (2016) datasets

4.5 Outlier-based SE results

In the section, we compare the different outlier-based SEs. Due to the small dimensionality of the Emmott et al. (2015) motherset anomaly benchmark datasets, we only used the sequential search methods to guide the search to compute the SE using an anomaly detector's outlier scoring measure. We excluded the PSO meta-heuristic search algorithm because the dimensionality of the Emmott et al. (2015) motherset anomaly benchmark datasets is small relative to the number of comparisons we make using the PSO meta-heuristic search. For example, the Magic Gamma motherset anomaly benchmark datasets have 11 features which are the most features from the Emmott et al. (2015) mothersets. A dimensionality of 11 features has a total of 2048 combinations. In our work, we restrict the total number of comparisons in the PSO meta-heuristic search to a maximum of 3000 per feature subset of size i in the SE $SE(\mathbf{x})_i$. Therefore, in place of the PSO meta-heuristic search algorithm, we use the AD Oracle SE to obtain the optimal subset of features to add in the SE based on the AD's outlier scoring measure. Due to the large amount of the Emmott et al. (2015) anomaly benchmark datasets from each motherset, we only computed the AD Oracle SEs for the Abalone, Concrete and Yeast motherset anomaly benchmarks because they have the smallest dimensionalities. For the Goldstein & Uchida (2016) anomaly benchmark datasets we computed the PSO meta-heuristic search algorithm because the dimensionality of the datasets is large and it would be computationally expensive and infeasible to compute the AD Oracle SE. For example, the Annthyroid anomaly benchmark dataset has the smallest dimensionality of 21 features. Therefore, just identifying the feature subspace that returns the best 6 features requires a total of $\binom{21}{6} = 54264$ computations. The PSO meta-heuristic search restricts the search space to a maximum of 3000 computations.

4.5.1 Comparison to AD Oracles

We observe from Table 4.1 that all of the AD Oracle SEs from the Abalone, Concrete and Yeast motherset anomaly benchmarks outperform their corresponding AD IND-FS, FS, IND-BS and BS SEs. These results show that a more extensive search in the search space yields improved results compared to using greedy sequential search methods. Also, the results show that there is a need for search methods that are not greedy and that a more extensive search improves the outlier-based SEs.

4.5.2 AD Oracle vs OptOracle

The OptOracle SE outperforms all of the AD Oracle SEs for the Abalone, Concrete and Yeast motherset anomaly benchmark datasets. From the analyst certainty curves in Figures 4.4 and 4.5 we observe that the gap between the best AD Oracle and OptOracle SEs for each motherset is significantly large. The gap shows that an exhaustive search across each AD's outlier scoring measure does not obtain the same SE as the OptOracle SE. The gap between the AD Oracle and OptOracle SEs could be due to the difference in the assumptions and properties of the anomaly detectors and OptOracle's notions of how to score outliers. In other words, there is a poor match between the OptOracle SE's notion of an outlier versus the anomaly detector's outlier scoring measure (Siddiqui et al. 2015, 2019).

4.5.3 Comparison to Random and OptOracle

OptOracle significantly outperforms all of the SEs, whereas, all of the SEs significantly outperform the Random SE with an exception to the Shuttle motherset EGMM and KDE sequential backward selection SEs. By analysing the analyst certainty curves of the Shuttle motherset's KDE and EGMM sequential backward selection SEs from Figure A.5 in Appendix A, we observe that the SEs start on a wrong note because the first feature in the SEs returns conditional probabilities that are lower than the first feature of the Random SE. The SEs only start to outperform the Random SE only after the sixth feature subset. Due to the greedy nature of KDE and EGMM sequential backward selection SEs, once a bad feature is chosen in the early stages of the SE it will remain in the subsequent feature subsets in the SE and affect the entire performance of the SE.

4.5.4 Comparison to PSO

Using the Goldstein & Uchida (2016) datasets, we compare the sequential search methods to the PSO meta-heuristic search method. From the AUCCs, we observe that given an AD's outlier scoring measure, the PSO meta-heuristic algorithm outperforms its corresponding sequential search SEs. The AUCCs show that the PSO meta-heuristic computes better SEs than the sequential search methods for the same AD scoring measure. The PSO meta-heuristic SE using an AD's outlier scoring measure can be outperformed by a sequential search method SE using a different AD outlier scoring measure. For example, the LOF PSO SE for the Aloi dataset has an average AUCC of 0.5489 and iTree FS has an average AUCC of 0.5705. Similar examples are observed throughout

the results in Table 4.3. The PSO meta-heuristic search only outperforms the sequential search methods that make use of the same AD outlier scoring measure and does not necessarily perform better than all the greedy sequential search methods that use different AD outlier scoring measures. The reason behind this could be because of the different assumptions and properties used by the different AD outlier scoring measures. A greedy sequential search outlier-based SE using an AD outlier scoring measure that measures the correct properties and uses the correct assumptions for a specific dataset will outperform a non-greedy PSO meta-heuristic SE using an AD outlier scoring measure that measures incorrect properties and uses the incorrect assumptions to score outliers on the dataset.

4.5.5 Sequential forward versus sequential backward selection SEs

In order to compare the sequential forward and backward selection search methods, we used a ranking measure to rank the methods in descending order of their AUCCs. For Tables 4.1 and 4.3 we ranked each row according to the sequential forward and backward selection search methods from 1 to 4 in descending order, then for each dataset we computed the mean rank for each search method. Lastly, we computed the mean of the mean ranks of each sequential backward and forward selection search method across all the datasets.

Overall, FS has the lowest mean rank followed by IND-FS, BS and IND-BS. From the mean ranks, it is clear that the sequential forward selection search SEs outperform the sequential backward selection search SEs. From analyst certainty curves from Appendices A and B, we observe that the first few feature subsets selected by the sequential backward selection search SEs return lower conditional probabilities than the features subsets that are selected by the sequential forward selection search methods. The sequential backward selection search SEs are therefore less robust in the early stages of the computation of the SEs. Siddiqui et al. (2015, 2019) found that a possible reason for the poor performance of outlier-based SEs guided by sequential backward selection search methods is that they produce a “weaker signal” compared to the SEs guided by sequential forward selection search methods in the early stages of the computation of the SEs. Siddiqui et al. (2015, 2019) found that when considering the outlyingness of single features, the differences in scores produced by sequential backward selection search methods for the features are often much smaller than the differences produced by the sequential forward selection search methods. Intuitively we would expect the BS search method to compute better explanations than the IND-FS search method because it takes feature interactions into account, however, because of the ”weak signal” that

the sequential backward selection search methods produce in the early stages of the computation of the SE, IND-FS was able to provide better explanations.

4.5.6 Comparison of AD outlier scoring measures

To identify the best AD outlier scoring measure, we rank the AD outlier scoring measures based on their AUCCs. For the Goldstein & Uchida (2016) datasets, we ranked the results of each outlier-based SE for each dataset from 1 to 36 using the AUCCs in Table 4.3. For each AD outlier scoring measure, we calculate its mean rank by calculating the mean of each row. Then for each AD outlier scoring measure, we computed the mean of the mean ranks across the Goldstein & Uchida (2016) datasets. Overall, iTree has the lowest mean ranking followed by EGMM, k-means, LOF, KDE and OCSVM. We obtained the same results for the Emmott et al. (2015) motherset datasets in Table 4.1, where we ranked the results along the columns of the sequential search methods.

4.6 Sample-based SE results

In this section, we analyse the results of the sample-based SEs. Similae to Section 4.5, due to the small dimensionality of the Emmott et al. (2015) datasets, we used all of the feature selection methods excluding PSO-SVM to compute the sample-based SEs. The AUCC results of the sample-based SEs can be found in Tables 4.2 and 4.4 for the Emmott et al. (2015) and Goldstein & Uchida (2016) datasets respectively, whereas the analyst certainty curves are in Column (b) of Figures 4.4, 4.5 and 4.6.

4.6.1 Comparison to Random and OptOracle

From the AUCC results and the analyst certainty curves, we observe that all of the sample-based SEs significantly outperform the Random SE. We observe from the Emmott et al. (2015) mothersets that the SEs significantly outperform the Random SE. However, there is a large gap between the sample-based SEs and the OptOracle SEs. The gap shows that there is a need for improved feature selection methods that are not greedy.

4.6.2 Comparison to PSO-SVM

From the AUCCs of the Goldstein & Uchida (2016) datasets we observe that the PSO-SVM SE outperforms the greedy feature selection based FS-SVM and BS-SVM SEs.

However, it does not always perform better than the other sample-based SEs. For example, in the Annthyroid, Aloi and KDD99 datasets, the SVM-RFE SE outperforms the PSO-SVM SE. A possible explanation for this would be that the PSO-SVM is a wrapper based feature selection method that relies on the accuracy of the SVM to select features. We observed in most cases that the SVM classifier was not able to obtain a 100% accuracy when splitting the inlier and outlier classes for the features that returned the highest classification accuracy, therefore resulting in sub-optimal solutions. The PSO-SVM SE improved on the greedy wrapper based FS-SVM and BS-SVM SEs only, and it fails to outperform the other SE methods consistently.

4.6.3 Comparison of feature selection methods

To obtain the best performing feature selection methods, we used a ranking measure to compare the different methods. To compute our ranking measure we did the following: for each of the results in Tables 4.2 and 4.4, we ranked each feature selection method in descending order based on their AUCCs for each dataset from 1 to 6 excluding the PSO-SVM in the Goldstein & Uchida (2016) datasets; we then averaged the rankings for each feature selection method across the datasets to obtain the overall mean rank. Overall, SVM-RFE had the lowest rank, followed by FS-SVM, RF-RFE, mRmR, MIFS and BS-SVM. Similarly to the sequential backward selection outlier-based SEs, the sampled-based BS-SVM SE also produces a "weak signal" in the early stages of the computation of the SE resulting in it to be the worst performing feature selection based method. Considering the Goldstein & Uchida (2016) dataset alone and the including PSO-SVM SE, we found that SVM-RFE had the lowest rank, followed by PSO-SVM, FS-SVM, mRmR, RF-RFE, MIFS and BS-SVM.

4.7 Computational complexity analysis

4.7.1 Complexity analysis of the outlier-based SEs

We now provide a computational complexity analysis of the different outlier-based SE methods. We chose to use time complexity analysis because it gives an estimation of the running time as a function of the size of the input data. Table 4.5 contains a summary of the complexity of the AD outlier scoring measures, while Table 4.6 contains a summary of the complexity of the search methods used to guide the AD outlier scoring measure.

TABLE 4.5: Complexity analysis of the anomaly detectors outlier scoring measures

AD scoring measure	**Complexity**
LOF	$O(nd + dn \log n)$
KDE	$O(nd)$
iTree	$O(Tn_t)$
***k*-means**	$O(nkdi)$
EGMM	$O(Lnkdi)$
OCSVM	$O(dn^3)$

The variable T for iTrees represents the number of trees, while n_t represents the sample size of t used to construct each tree from the original full dataset of size n. The variable i represents the total number of iterations and the variable L for the EGMM represents the total number of GMMs that will be fitted.

TABLE 4.6: Complexity analysis of the search methods

Search method	**Complexity**
IND-FS	$O(d)$
FS	$O(d^2)$
IND-BS	$O(d)$
BS	$O(d^2)$
PSO	$O(GPd_{\max})$

The variable $d_{\max}$ is the total number of feature subsets in the SE, G represents the number of generations of the PSO, and P is the total number of particles.

4.7.2 Complexity of the sample-based SEs

The most significant component of the sampled-based SE's complexity is the selection of the k-nearest neighbours required to construct the inlier class. We used KDTrees (Bhatia et al. 2010) for the nearest neighbour search, which has a complexity of $O(dn \log n)$. We represent the total number of data points in the inlier and outlier class by n_s. The filter-based feature selection methods MIFS and mRmR have a complexity of $O(d)$ computations. The SVM model used by the the wrapper-based methods FS-SVM and BS-SVM SEs has a complexity of $O(dn_s^3)$ and these SEs are guided by search methods with a complexities of $O(d^2)$ computations while the PSO-SVM SE is guided by a meta-hueristic with a complexity of $O(GPd_{\max})$ computations. The embedded-based feature

selection methods SVM-RFE and RF-RFE SEs are guided by a search method that takes $O(d^2)$ computations. The RF model has a complexity of $O(Tdn_s \log n_s)$, where T represents the total number of trees.

4.8 Results discussion

In this section, compare the results of the sample and outlier based SE, briefly discuss the possible trade-offs an analyst could face between choosing SEs that return quality explanations and SEs that are fast to compute and lastly, we explicitly provide answers to our research questions in Section 1.4 of Chapter 1.

4.8.1 Sample-based versus outlier-based SEs

For the Emmott et al. (2015) datasets, we used the same ranking method as in Section 4.6.3 and ranked the AUCCs of the outlier and sampled-based SEs from Tables 4.1 and 4.2 respectively and obtained the following rankings: SVM-RFE had the lowest rank, followed by FS-SVM, RF-RFE, the iTree SEs, mRmR, MIFS, BS-SVM, the EGMM SEs, the LOF SEs, the k-means SEs and the OCSVM SEs. For the Goldstein & Uchida (2016) datasets, we ranked the AUCCs of the outlier and sample-based SEs from Tables 4.3 and 4.4 respectively and obtained the following rankings: SVM-RFE had the lowest rank, followed by PSO-SVM, the iTree SEs, FS-SVM, RF-RFE, mRmR, MIFS, BS-SVM, the EGMM SEs, the LOF SEs, the k-means SEs and the OCSVM SEs. It is clear from these results that the sample-based SEs outperform the outlier-based SEs in terms of AUCCs.

4.8.2 Trade off between AUCCs and computational complexity

Some SEs may produce good results, but they may also take longer to compute the required features into the SEs. The analyst could then choose to trade off a good performing SE for alternative SEs that are computationally cheaper to compute even though they do return the best results. For example, our results have shown that the sample-based SVM-RFE is the best performing SE, while the iTree AD is the best AD to use to compute outlier based SEs. SVM-RFE SE is computationally more expensive than to compute than the iTree sequential search SEs. Based on the analysts time constraints, they might consider computing an iTree sequential search SE over the SVM-RFE SE in order to trade performance and meet the time constraints they face.

4.8.3 Answers to the research questions

1. Outlier-based SE research questions:

   (a) **Do the outlier-based SEs provide explanations that are better than the analyst manually browsing for the outlying feature subsets?**

   With an exception to the Shuttle motherset EGMM and KDE sequential backward selection SEs, all of the SEs from work outperformed the Random SE. We identified that the possible reason for the poor performance of the Shuttle motherset EGMM and KDE sequential backward selection SEs could be because of the weak signal that sequential backward selection SEs produce. Overall, the outlier-based SEs provide the analyst with a better explanation than manually browsing for the outlying feature subsets.

   (b) **Which anomaly detector's outlier scoring measure computes the best outlier-based SEs?**

   The iTree anomaly detector's outlier scoring measure computes the best outlier-based SEs.

   (c) **Between the greedy nested sequential search methods and the non-nested meta-heuristic search method, which one computes the best outlier-based SEs?**

   The PSO meta-heuristic search method computed the best performing outlier-based SEs.

   (d) **Which anomaly detector's outlier scoring measure paired with a search method computes the best outlier-based SE?**

   The iTree PSO SE computes the best performing outlier-based SE.

   (e) **What is the complexity of the different anomaly detector's outlier scoring measures and search methods?**

   The complexity analysis can be found in Section 4.7.1

2. Sample-based SE research questions:

   (a) **Do the sample-based SEs provide explanations that are better than the analyst manually browsing for the outlying feature subsets?**

   All of the sample-based SEs significantly outperformed the random SE. Therefore, the sample-based SEs provide the analyst with better explanations than manually browsing for the outlying feature subsets.

   (b) **Which feature selection method computes the best sample-based SEs?**

   SVM-RFE computes the best sample-based SEs.

(c) **Between the greedy nested sequential search methods and the non-nested meta-heuristic search method, which one computes the best sample-based SEs?**

The SVM-RFE SE which uses a greedy nested sequential search method computes the best performing SE and outperforms the PSO-SVM SE which is guided by a meta-heuristic search method.

(d) **What is the complexity of the different feature selection methods?**

The complexity analysis can be found in Section 4.7.2

3. **Between the sample and outlier-based SEs, which method computes the best SEs?**

   The sample-based SEs compute the best performing SEs

4. **Is the AUCC evaluation measure better than the MFP evaluation measure?**

   The AUCC evaluation measure provides a better evaluation method than the MFP. Also, the analyst certainty curves give us more insight into the performance of every feature subset in the SE.

4.9 Conclusion

In this chapter, we analysed the results of our experiments. In Section 4.2, we presented the Random and Oracle SEs, which respectively provide lower and upper bound performance thresholds for the SEs. We found that the sample and outlier-based SEs significantly outperformed the Random SEs. However, there is still a gap between the OptOracle SEs and sample and outlier-based SE. In particular, the gap between the AD Oracle and OptOracle is still significantly large even though the AD Oracle is an exhaustive search using an AD outlier scoring measure. The gap between the AD Oracle and OptOracle SEs could be because of a poor match between the OptOracle's notion of an outlier versus the anomaly detector's notion of an outlier (Siddiqui et al. 2015, 2019).

In Section 4.3, we analysed the results of the sample and outlier-based SEs using the AUCCs and the analyst certainty curves. We observed that using the AUCCs alone to compare the SEs is not enough. Using the AUCCs, along with the analyst certainty curves, gives us more insight into the overall performance of the SEs. In Section 4.4, we compared the MFP evaluation method to the AUCCs. We found that the AUCCs, along with the analyst certainty curves, provide a better evaluation method than using the MFP.

Section 4.5 presents the outlier-based SE results. We found that the iTree AD outlier measure computes the best performing SEs. The PSO meta-heuristic search only outperforms the greedy search methods that make use of the same AD outlier scoring measure and does not necessarily perform better than all of the greedy explanation methods that are using different AD outlier scoring measures for their SEs. From the sequential search methods, FS provided the best SEs followed by IND-FS, BS and IND-BS. The bad performance of the sequential backward selection search methods could be because of the "weaker signals" that they produce during the computation of the SEs. We would expect BS to provide better explanations than IND-FS because it takes feature interactions into account, however, because of the "weak signals" that they produce in the early stages of the computation of the SE, IND-FS computed better explanations.

Section 4.6 presents the sample-based SE results. The SVM-RFE feature selection method computed the best SEs and also outperformed the PSO-SVM. The PSO-SVM only improved on the greedy wrapper-based FS-SVM and BS-SVM SEs and failed to outperform the other SE methods consistently. A possible reason that SVM-RFE outperformed PSO-SVM could be because the classifier was not able to fully separate the inlier and outlier classes, therefore, resulting in sub-optimal SEs.

Lastly Sections 4.7 and 4.8 respectively provide the complexity analysis and final discussion of the results. Overall, we found that the sample-based SVM-RFE provided the best SEs and that the sample-based SEs computed better SEs than the outlier-based SEs.

Chapter 5

Conclusion

Anomaly detection is the problem of identifying the process that generates anomalies. However, in most cases the process generating anomalies is unknown, therefore, resulting in anomaly detection methods in real-world applications to operate in an unsupervised mode by looking for outliers in the dataset with the hope that they are anomalies (Siddiqui et al. 2015, 2019, Emmott et al. 2015). An analyst is then faced with the challenge of examining all of the outliers from the anomaly detector to identify the features associated with each outlier that make the outlier outlying to decide whether it is an anomaly or not. In most cases, the analyst has to manually browse through the full feature space of each outlier point to identify the outlying subset of features that will give them enough information to classify whether the outlier is truly anomalous or not. The identification of the outlying subset of features is an extremely challenging task, even if tens of features characterise the feature space. Identifying the outlying subsets of features becomes more complicated when feature interactions are also critical in decision making.

Our study aims to ease the load on the analysts by providing them with SEs that will provide them with explanations on which features make an outlier point outlying and to also improve the evaluation methods of SEs. In our work, we developed two novel approaches of developing SEs named the outlier and sample-based SEs. The outlier-based SEs use an anomaly detector's outlier scoring measure guided by search methods to compute SEs. The sample-based SEs turn the problem into a feature selection problem and make use of feature selection methods to compute SEs. Also, we developed a new evaluation method, where we measure the AUCCs of the SEs and compare them based on the AUCCs along with the analyst certainty curves.

We were able to show the following through our study:

- Both the outlier and sample-based SEs were able to provide the analyst with a better explanation than manually browsing for the outlying feature subsets.
- The iTree anomaly detector's outlier scoring measure computes the best outlier-based SEs while the SVM-RFE sample-based SE computes the best performing sample-based SEs.
- The non-nested PSO meta-heuristic search method computes the best performing outlier-based SEs; however, it only outperforms the greedy search methods that make use of the same AD outlier scoring measure and does not necessarily perform better than all of the greedy sequential search explanation methods that are using different AD outlier scoring measures. Whereas, for sample-based SEs, the PSO-SVM SE is outperformed by the greedy SVM-RFE. We identified that the reason SVM-RFE outperforms the PSO-SVM SE is that the classifier for the PSO-SVM RFE was not able to obtain a 100% accuracy when splitting the inlier and outlier classes for the features that returned the highest classification accuracy, therefore resulting in sub-optimal solutions.
- The sample-based SEs compute the best performing SEs.
- The AUCC evaluation measure, along with analysing the analyst certainty curves provide a better evaluation method than the MFP.
- Even though the sample-based SVM-RFE is the best performing SE, it is computationally more expensive than to compute than the outlier-based iTree sequential search SEs which are the computationally the cheapest SEs to compute. Therefore, based on the analysts time constraints, they might consider computing one of the iTrees sequential search SEs over the SVM-RFE SE in order to trade performance and meet the time constraints they face.

In conclusion the sample and outlier-based SEs can be used alongside any anomaly detection algorithm to explain what makes a data point an anomaly to an analyst. In addition, our new evaluation method the AUCC is a significant improvement on the existing evaluation methods of evaluating explanations.

Appendix A

Emmott et al. (2015)

A.1 Abalone

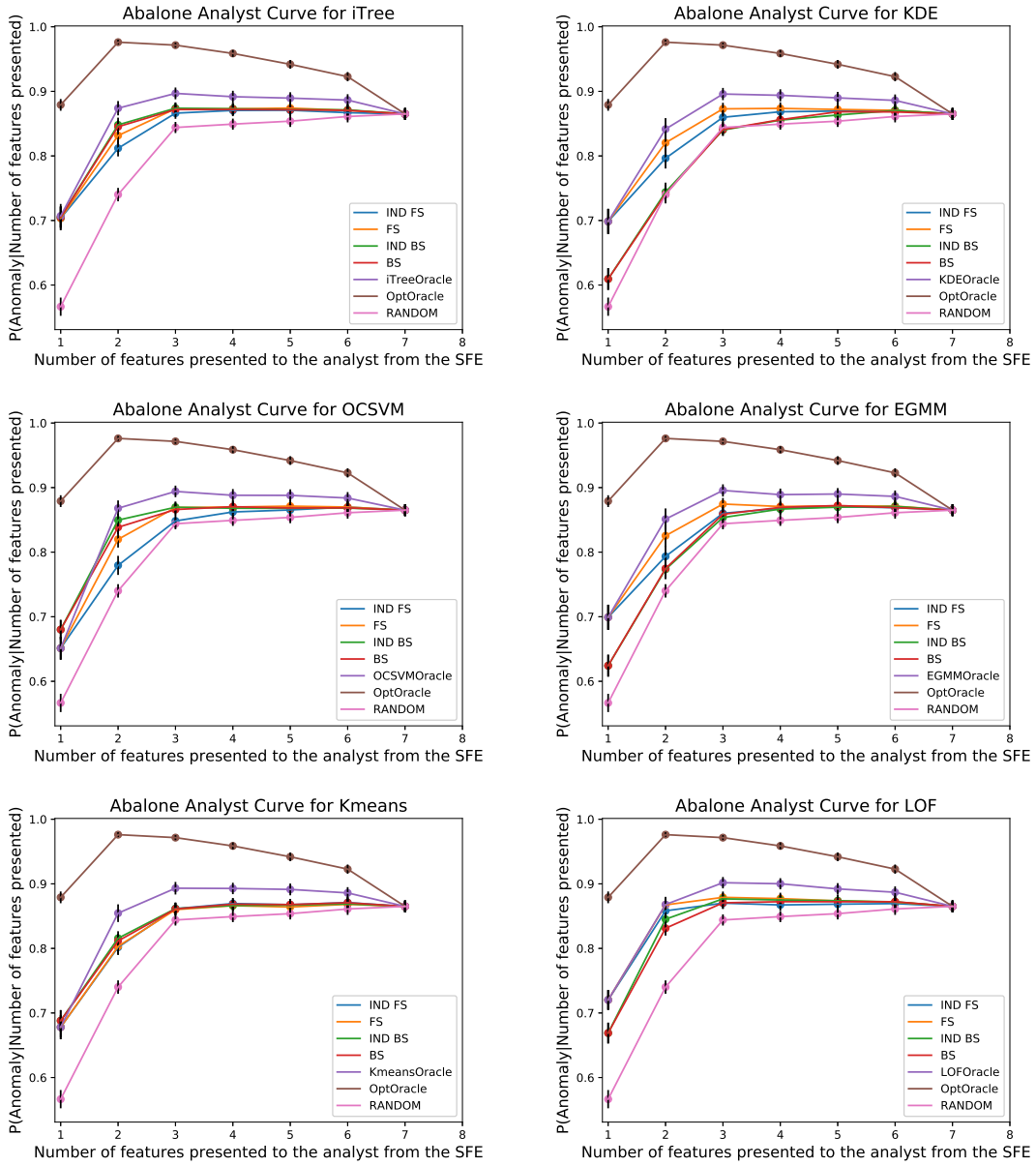


FIGURE A.1: Abalone outlier-based SE analyst certainty curves

A.2 Concrete

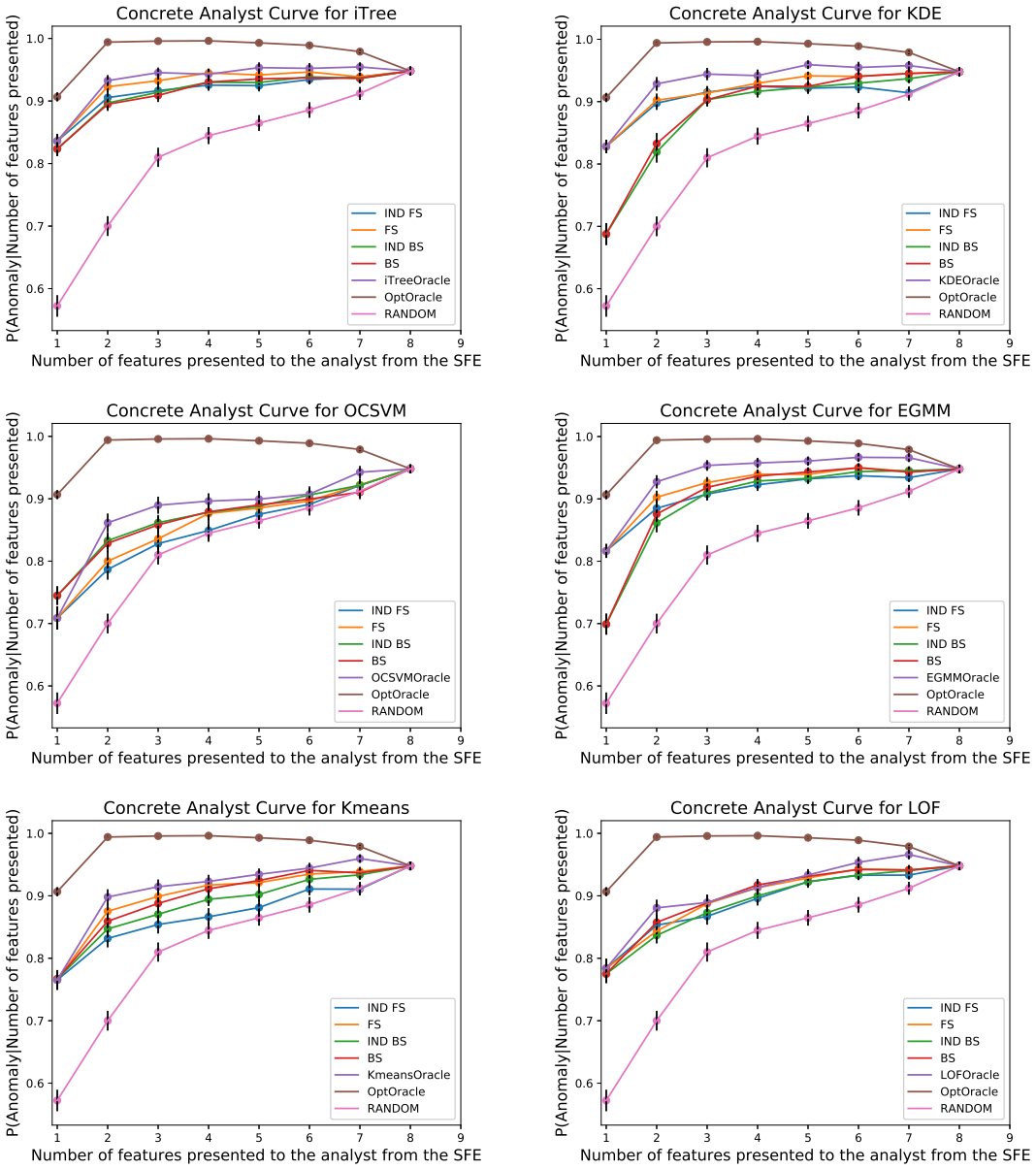


FIGURE A.2: Concrete outlier-based SE analyst certainty curves

A.3 Magic Gamma

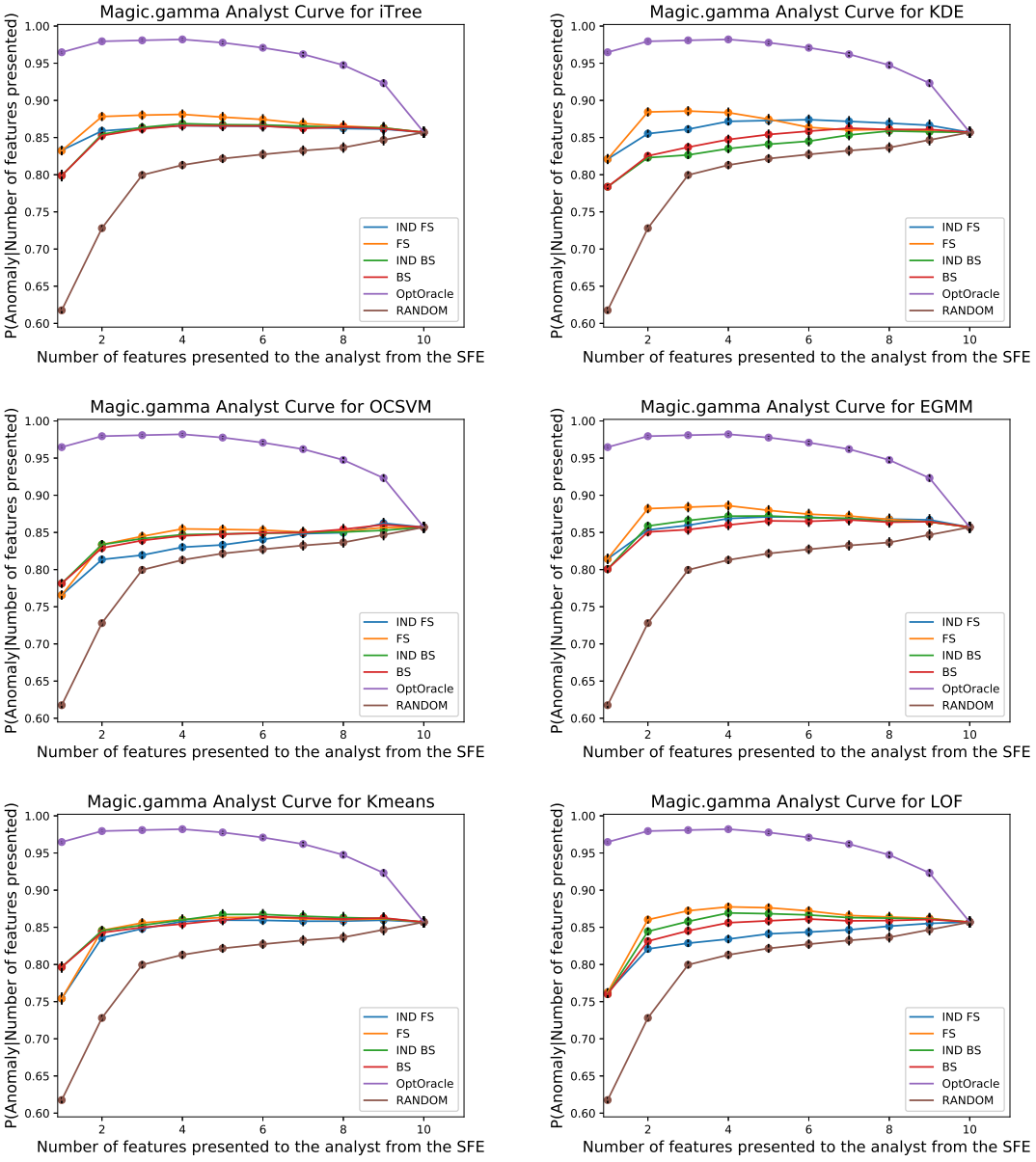


FIGURE A.3: Magic Gamma outlier-based SE analyst certainty curves

A.4 Pageb

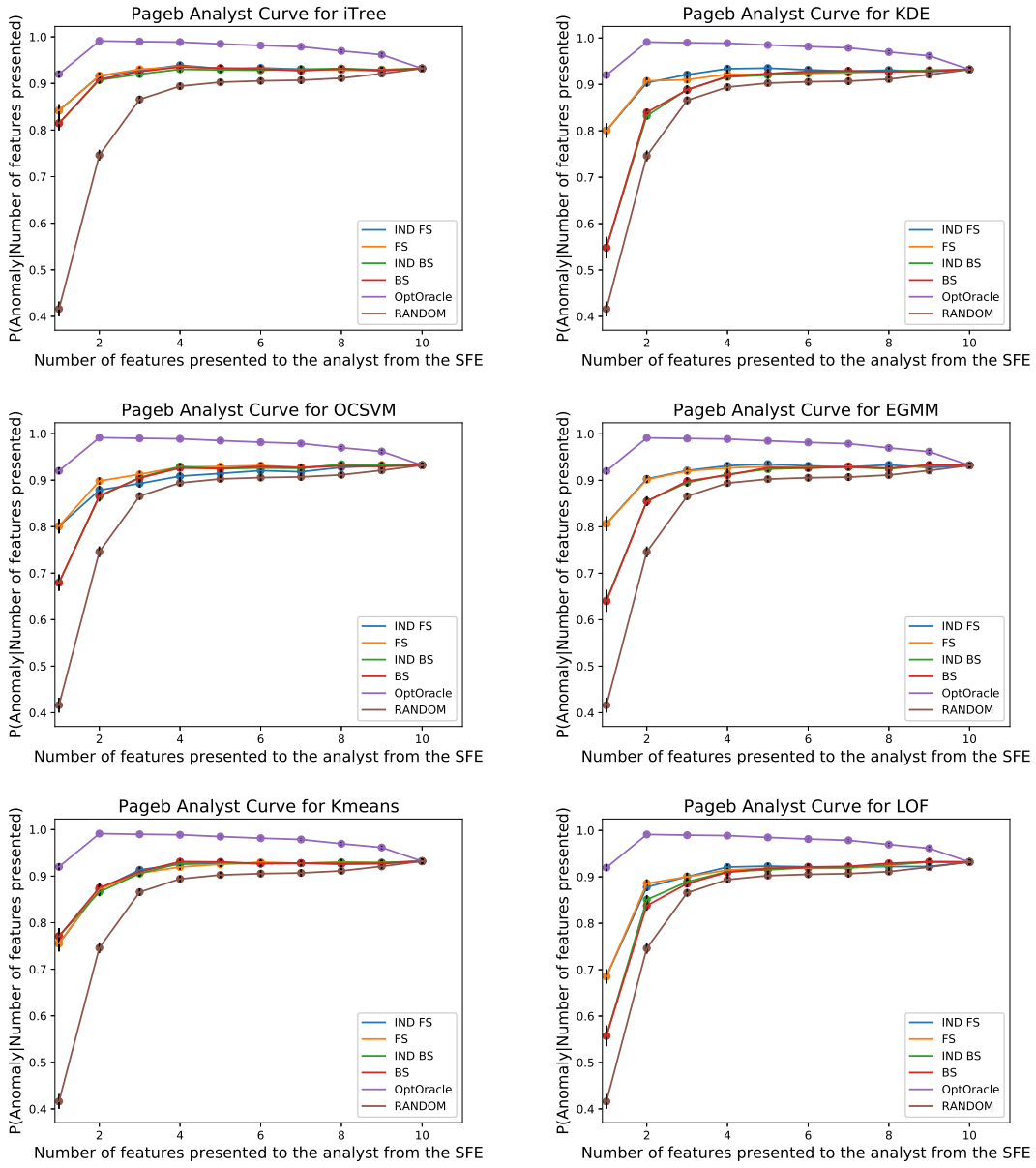


FIGURE A.4: Pageb outlier-based SE analyst certainty curves

A.5 Shuttle

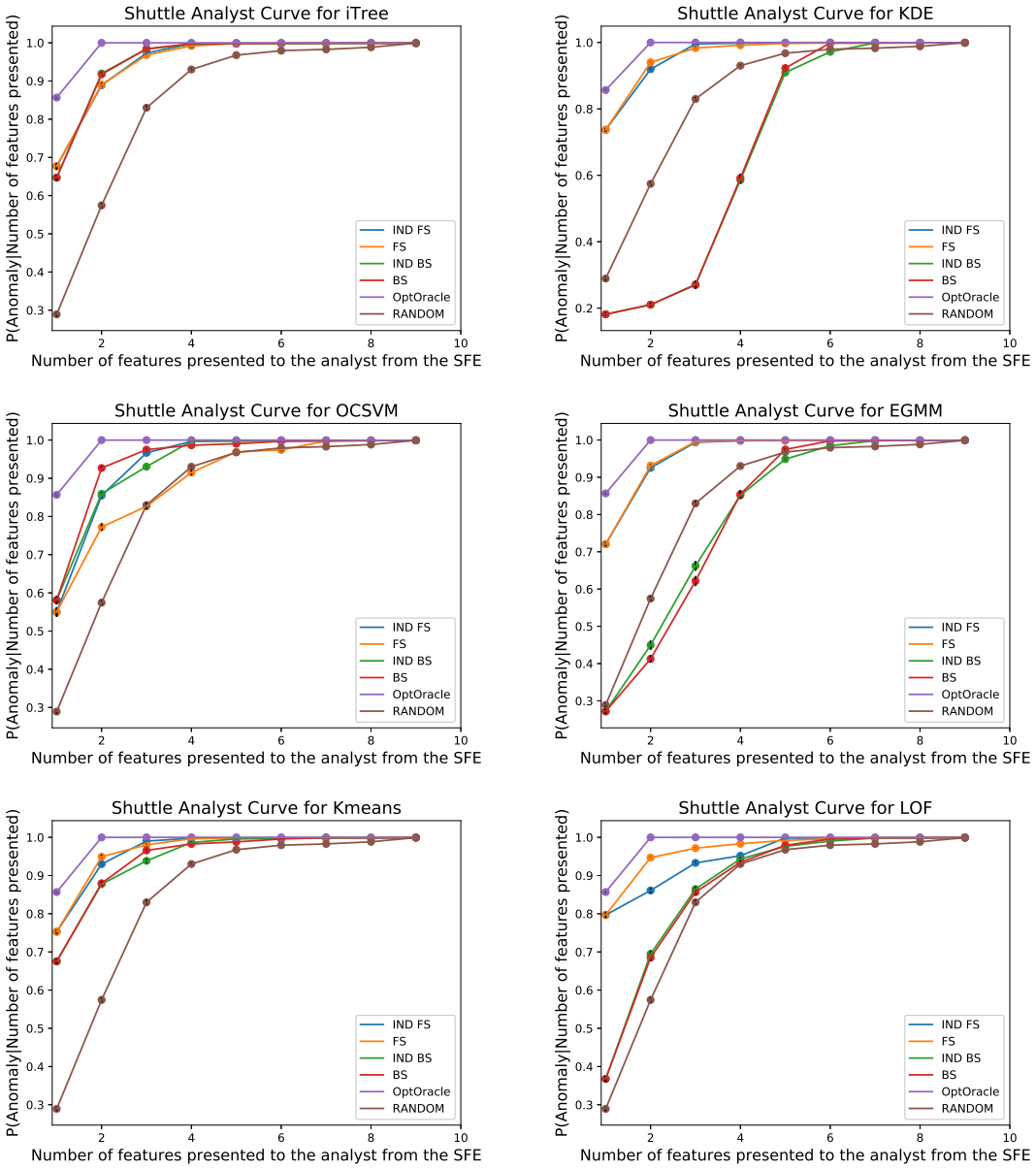


FIGURE A.5: Shuttle outlier-based SE analyst certainty curves

A.6 Wine

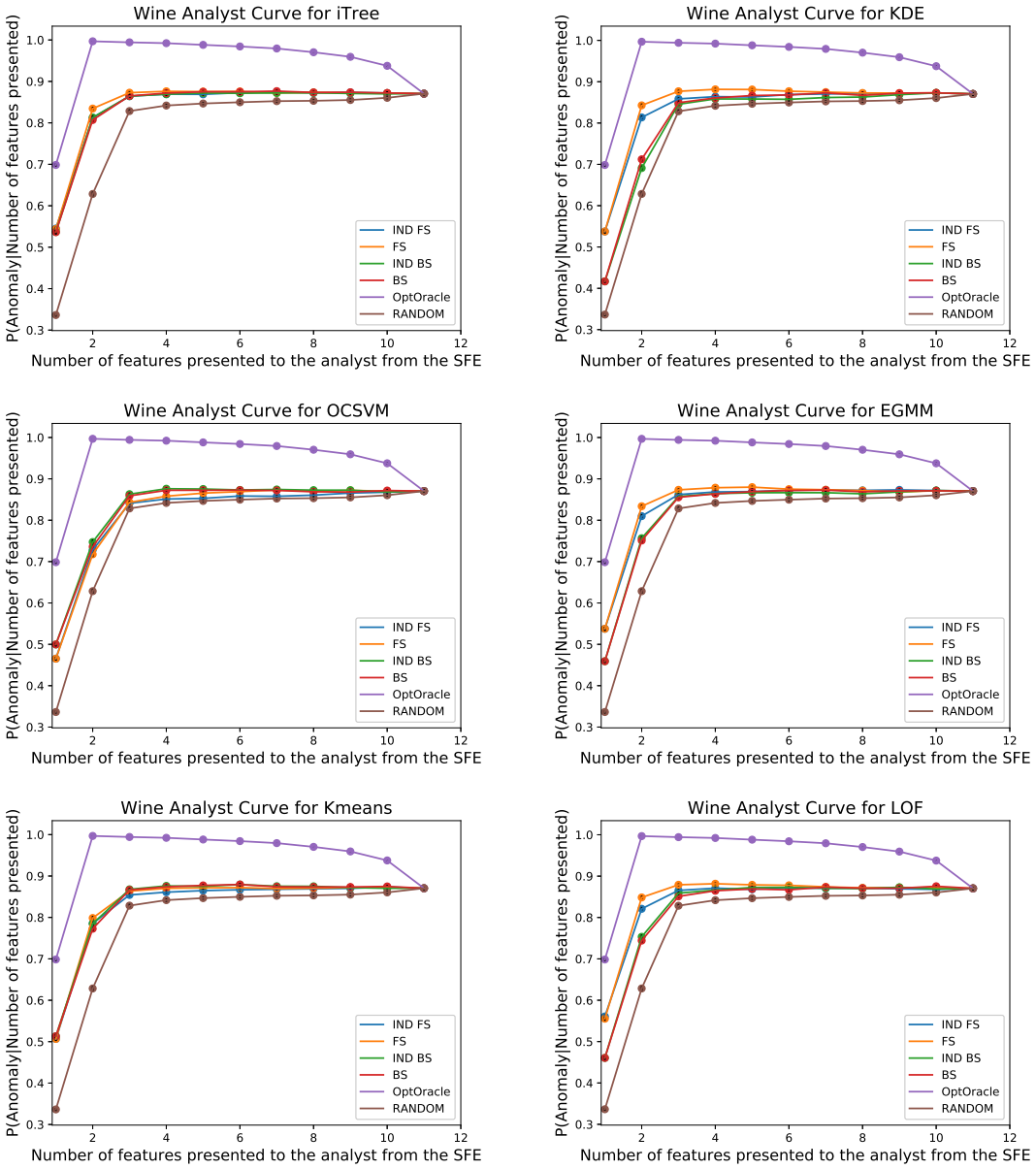


FIGURE A.6: Wine outlier-based SE analyst certainty curves

A.7 Yeast

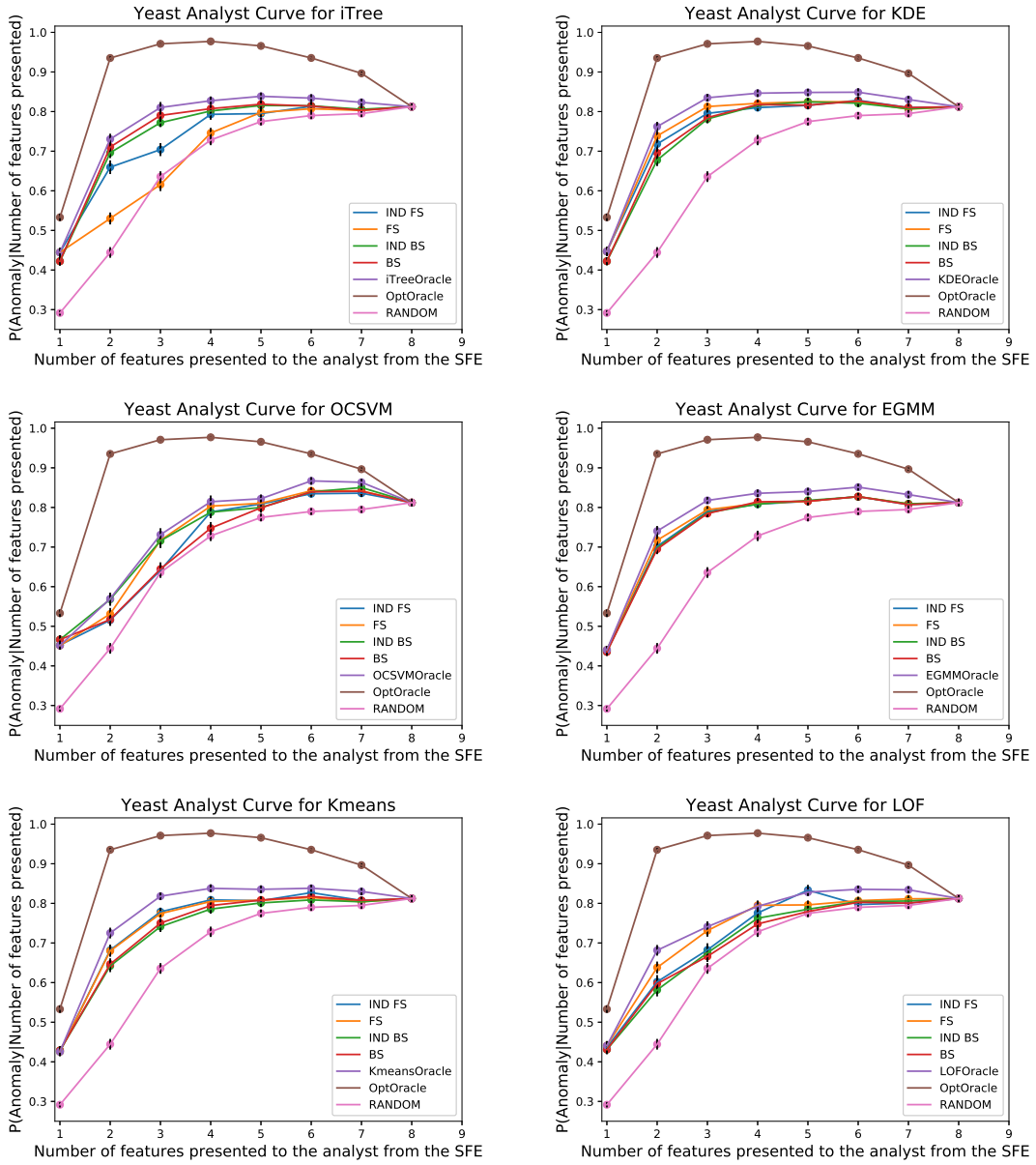


FIGURE A.7: Yeast outlier-based SE analyst certainty curves

Appendix B

Goldstein & Uchida (2016)

B.1 Aloi

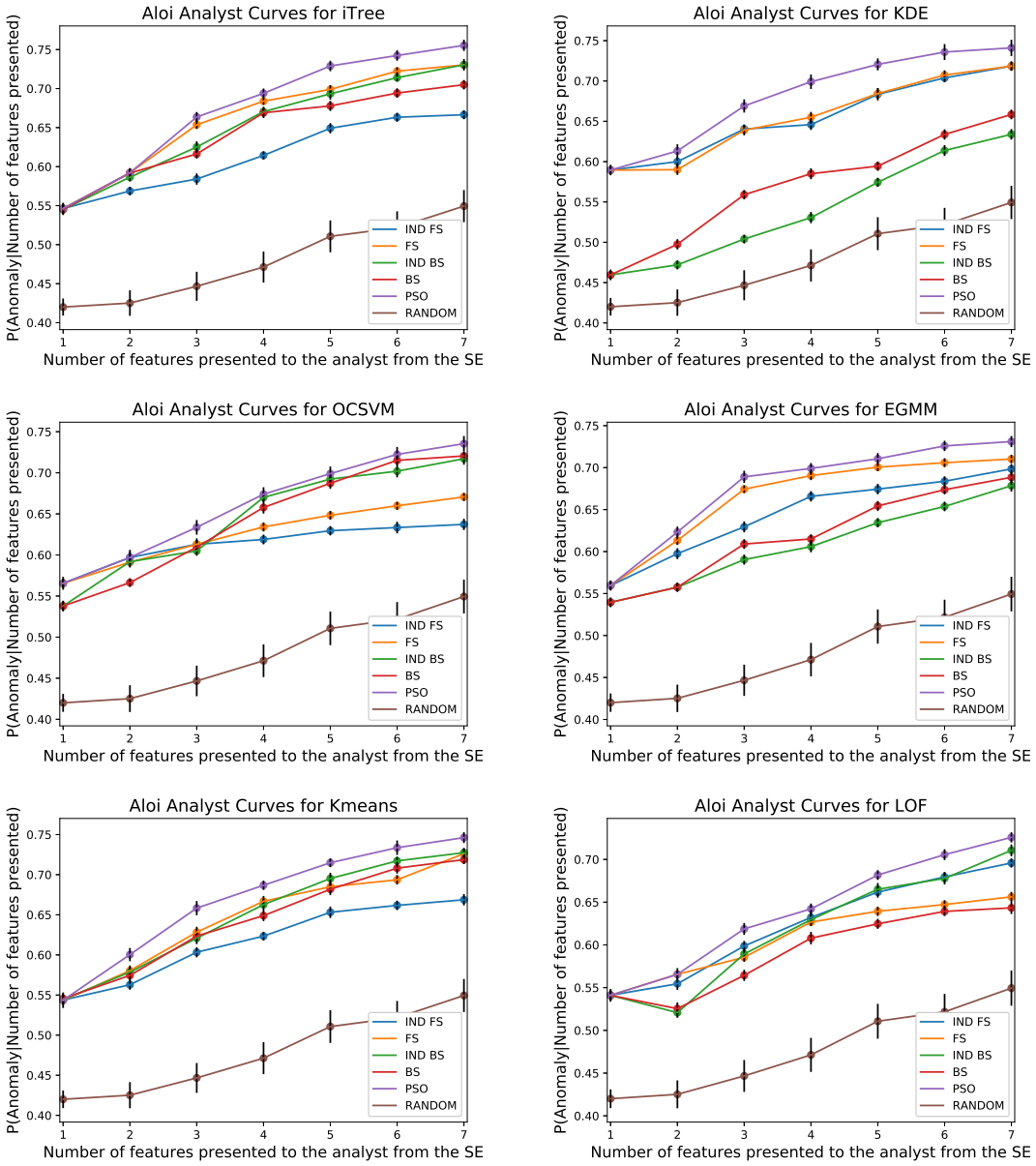


FIGURE B.1: Aloi outlier-based SE analyst certainty curves

B.2 Annthyroid

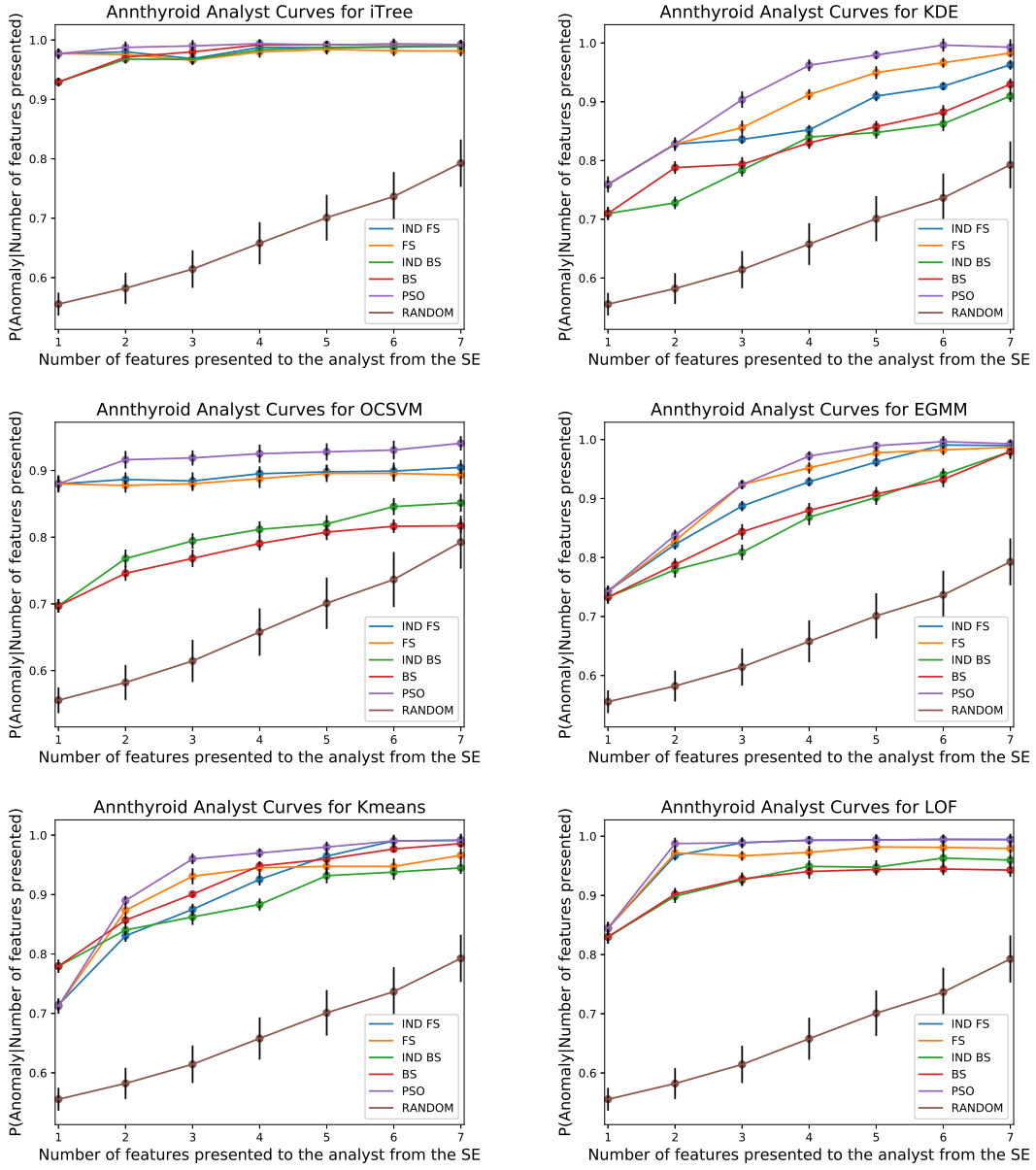


FIGURE B.2: Annthyroid outlier-based SE analyst certainty curves

B.3 KDD99

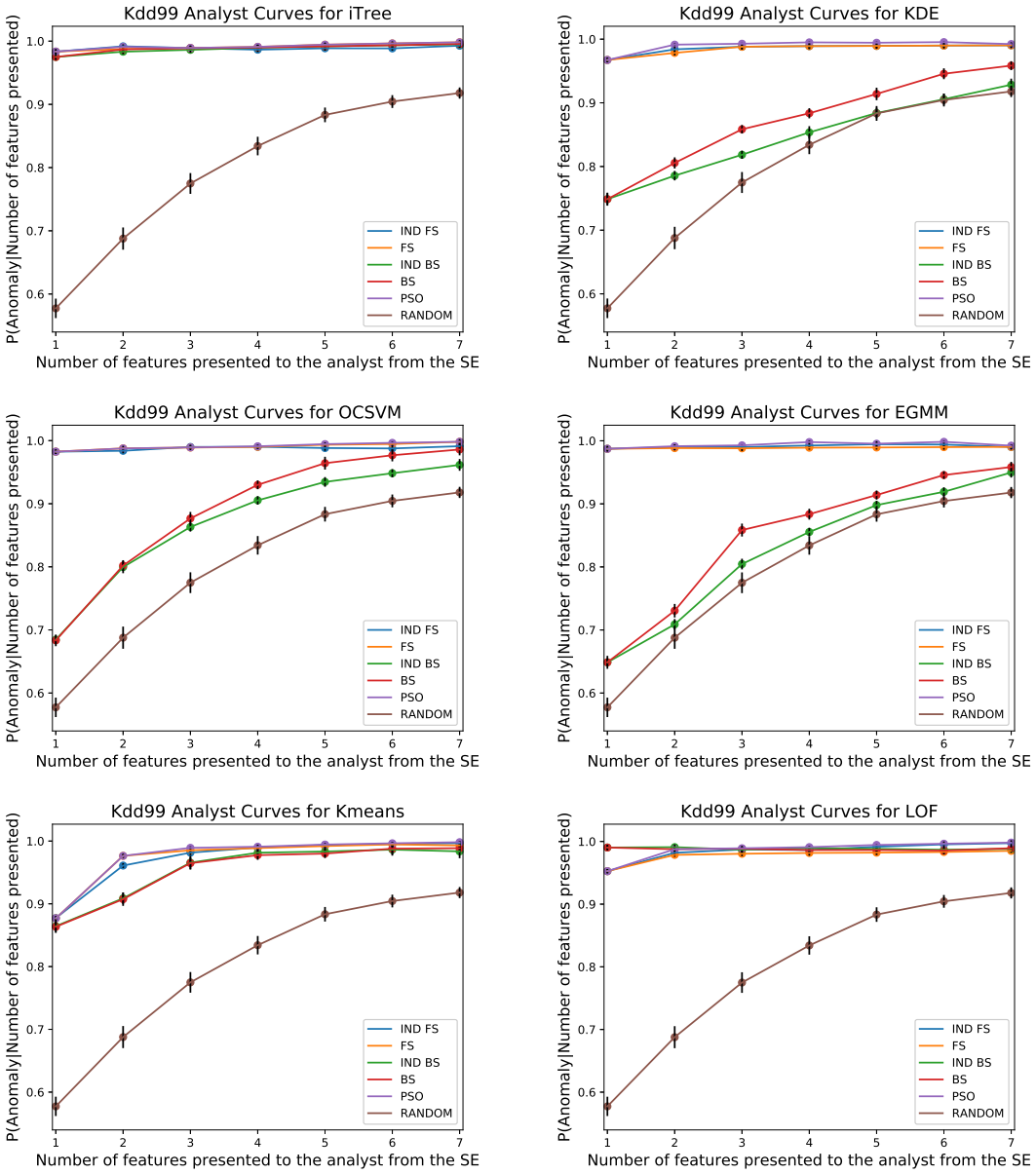


FIGURE B.3: KDD99 outlier-based SE analyst certainty curves

B.4 Satellite

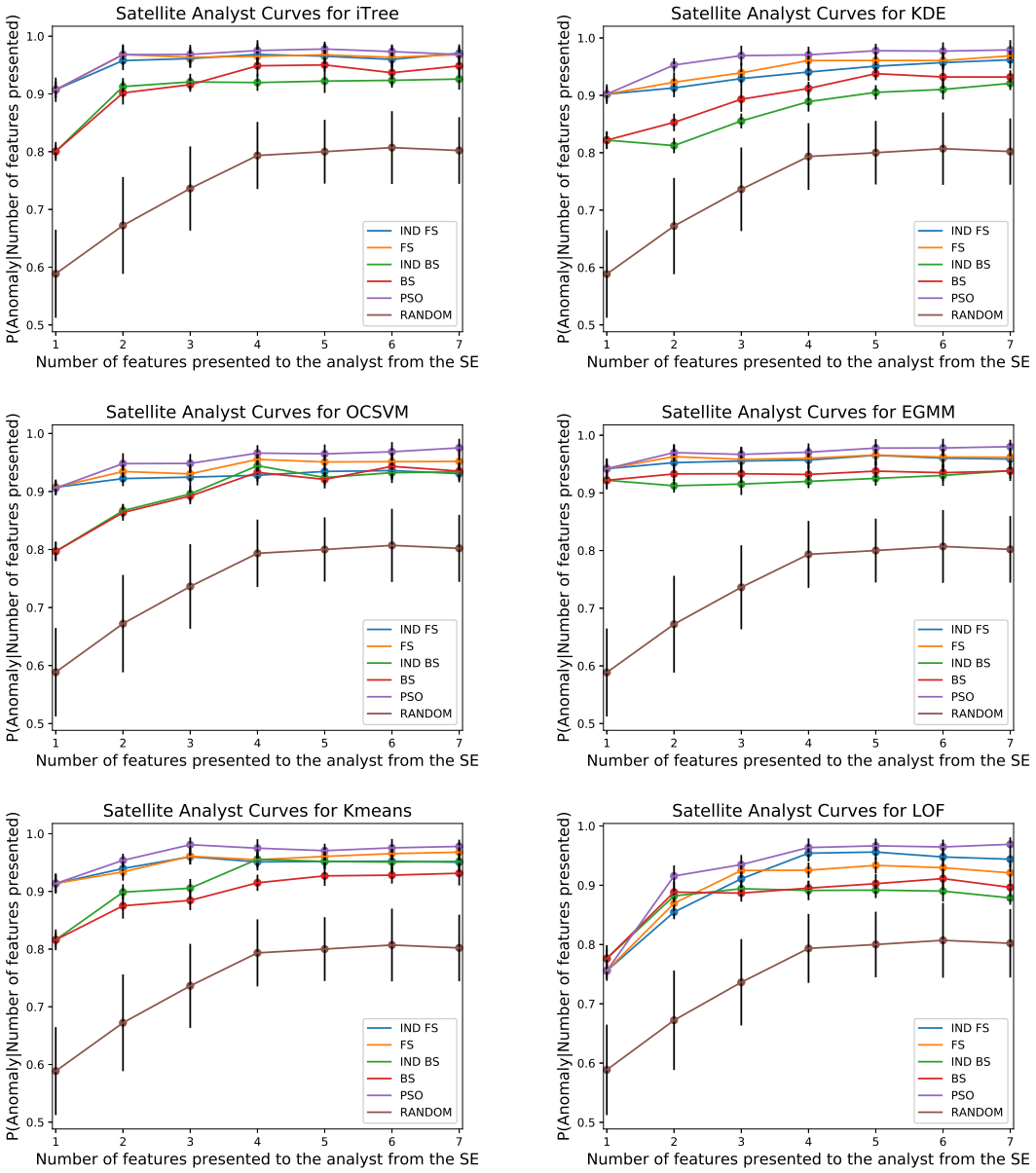


FIGURE B.4: Satellite outlier-based SE analyst certainty curves

Bibliography

Alzghoul, A. & Löfstrand, M. (2011), 'Increasing availability of industrial systems through data stream mining', *Computers & Industrial Engineering* **60**(2), 195–205.

Anantharam, P., Thirunarayan, K. & Sheth, A. (2012), Topical anomaly detection from twitter stream, *in* 'Proceedings of the 4th Annual ACM Web Science Conference', WebSci '12, ACM, New York, NY, USA, pp. 11–14.
URL: *http://doi.acm.org/10.1145/2380718.2380720*

Angiulli, F., Fassetti, F., Manco, G. & Palopoli, L. (2017), 'Outlying property detection with numerical attributes', *Data mining and knowledge discovery* **31**(1), 134–163.

Angiulli, F., Fassetti, F. & Palopoli, L. (2009), 'Detecting outlying properties of exceptional objects', *ACM Transactions on Database Systems (TODS)* **34**(1), 7.

Angiulli, F., Fassetti, F. & Palopoli, L. (2013), 'Discovering characterizations of the behavior of anomalous subpopulations', *IEEE Transactions on knowledge and data engineering* **25**(6), 1280–1292.

Basu, S. & Meckesheimer, M. (2007), 'Automatic outlier detection for time series: an application to sensor data', *Knowledge and Information Systems* **11**(2), 137–154.

Battiti, R. (1994), 'Using mutual information for selecting features in supervised neural net learning', *IEEE Transactions on neural networks* **5**(4), 537–550.

Bergstra, J. & Bengio, Y. (2012), 'Random search for hyper-parameter optimization', *Journal of Machine Learning Research* **13**(Feb), 281–305.

Bhatia, N. et al. (2010), 'Survey of nearest neighbor techniques', *arXiv preprint arXiv:1007.0085* .

Bishop, C. (2007), 'Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn'.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. & Sander, J. (2000), Lof: identifying density-based local outliers, *in* 'ACM sigmod record', Vol. 29, ACM, pp. 93–104.

Brown, G., Pocock, A., Zhao, M.-J. & Luján, M. (2012), 'Conditional likelihood maximisation: a unifying framework for information theoretic feature selection', *Journal of machine learning research* **13**(Jan), 27–66.

Chandola, V., Banerjee, A. & Kumar, V. (2009), 'Anomaly detection: A survey', *ACM computing surveys (CSUR)* **41**(3), 15.

Chandrashekar, G. & Sahin, F. (2014), 'A survey on feature selection methods', *Computers & Electrical Engineering* **40**(1), 16–28.

Dang, X. H., Assent, I., Ng, R. T., Zimek, A. & Schubert, E. (2014), Discriminative features for identifying and interpreting outliers, *in* '2014 IEEE 30th International Conference on Data Engineering', IEEE, pp. 88–99.

Dang, X. H., Micenková, B., Assent, I. & Ng, R. T. (2013), Local outlier detection with interpretation, *in* 'Joint European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 304–320.

Dorigo, M. & Di Caro, G. (1999), Ant colony optimization: a new meta-heuristic, *in* 'Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)', Vol. 2, IEEE, pp. 1470–1477.

Du, W., Fang, L. & Peng, N. (2006), 'Lad: Localization anomaly detection for wireless sensor networks', *Journal of parallel and distributed computing* **66**(7), 874–886.

Dua, D. & Karra Taniskidou, E. (2017), 'UCI machine learning repository'.
URL: *http://archive.ics.uci.edu/ml*

Duan, L., Tang, G., Pei, J., Bailey, J., Campbell, A. & Tang, C. (2015), 'Mining outlying aspects on numeric data', *Data Mining and Knowledge Discovery* **29**(5), 1116–1151.

Emmott, A., Das, S., Dietterich, T., Fern, A. & Wong, W.-K. (2015), 'A meta-analysis of the anomaly detection problem', *arXiv preprint arXiv:1503.01158* .

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G. & Vázquez, E. (2009), 'Anomaly-based network intrusion detection: Techniques, systems and challenges', *computers & security* **28**(1), 18–28.

Geusebroek, J.-M., Burghouts, G. J. & Smeulders, A. W. (2005), 'The amsterdam library of object images', *International Journal of Computer Vision* **61**(1), 103–112.

Glodek, M., Schels, M. & Schwenker, F. (2013), 'Ensemble gaussian mixture models for probability density estimation', *Computational Statistics* **28**(1), 127–138.

Goldstein, M. & Uchida, S. (2016), 'A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data', *PloS one* **11**(4), e0152173.

Greensmith, J., Twycross, J. & Aickelin, U. (2006), Dendritic cells for anomaly detection, *in* 'Evolutionary Computation, 2006. CEC 2006. IEEE Congress on', IEEE, pp. 664–671.

Gupta, N., Eswaran, D., Shah, N., Akoglu, L. & Faloutsos, C. (2018), Beyond outlier detection: Lookout for pictorial explanation, *in* 'Joint European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 122–138.

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002), 'Gene selection for cancer classification using support vector machines', *Machine learning* **46**(1-3), 389–422.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H. & Bing, G. (2017), 'Learning from class-imbalanced data: Review of methods and applications', *Expert Systems with Applications* **73**, 220–239.

Hanley, J. A. & McNeil, B. J. (1982), 'The meaning and use of the area under a receiver operating characteristic (roc) curve.', *Radiology* **143**(1), 29–36.

Hawkins, S., He, H., Williams, G. & Baxter, R. (2002), Outlier detection using replicator neural networks, *in* 'International Conference on Data Warehousing and Knowledge Discovery', Springer, pp. 170–180.

Hsu, C.-W., Chang, C.-C., Lin, C.-J. et al. (2003), 'A practical guide to support vector classification'.

Hussain, K., Salleh, M. N. M., Cheng, S. & Shi, Y. (2018), 'Metaheuristic research: a comprehensive survey', *Artificial Intelligence Review* pp. 1–43.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R. & Wu, A. Y. (2002), 'An efficient k-means clustering algorithm: Analysis and implementation', *IEEE Transactions on Pattern Analysis & Machine Intelligence* (7), 881–892.

Kennedy, J. (2010), 'Particle swarm optimization', *Encyclopedia of machine learning* pp. 760–766.

Kirkos, E., Spathis, C. & Manolopoulos, Y. (2007), 'Data mining techniques for the detection of fraudulent financial statements', *Expert systems with applications* **32**(4), 995–1003.

Kohavi, R. et al. (1995), 'A study of cross-validation and bootstrap for accuracy estimation and model selection', **14**(2), 1137–1145.

Kopp, M., Pevnỳ, T. & Holena, M. (2014), Interpreting and clustering outliers with sapling random forests, *in* 'Information Technologies Applications and Theory Workshops, Posters, and Tutorials (ITAT)'.

Krawczyk, B. (2016), 'Learning from imbalanced data: open challenges and future directions', *Progress in Artificial Intelligence* **5**(4), 221–232.

Kriegel, H.-P., Kröger, P., Schubert, E. & Zimek, A. (2009), Outlier detection in axis-parallel subspaces of high dimensional data, *in* 'Pacific-Asia Conference on Knowledge Discovery and Data Mining', Springer, pp. 831–838.

Kriegel, H.-P., Kröger, P., Schubert, E. & Zimek, A. (2012), Outlier detection in arbitrarily oriented subspaces, *in* '2012 IEEE 12th international conference on data mining', IEEE, pp. 379–388.

Kriegel, H.-P., Zimek, A. et al. (2008), Angle-based outlier detection in high-dimensional data, *in* 'Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 444–452.

Kuo, C.-T. & Davidson, I. (2016), A framework for outlier description using constraint programming, *in* 'Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence', AAAI Press, pp. 1237–1243.

Laorden, C., Ugarte-Pedrero, X., Santos, I., Sanz, B., Nieves, J. & Bringas, P. G. (2014), 'Study on the effectiveness of anomaly detection for spam filtering', *Information Sciences* **277**, 421–444.

Lee, W., Stolfo, S. J. et al. (1998), Data mining approaches for intrusion detection., *in* 'Usenix security'.

Lewis, R. J. (2000), An introduction to classification and regression tree (cart) analysis, *in* 'Annual meeting of the society for academic emergency medicine in San Francisco, California', Vol. 14.

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

Liaw, A. & Wiener, M. (2002), 'Classification and regression by randomforest', *R news* **2**(3), 18–22.

Liu, F. T., Ting, K. M. & Zhou, Z.-H. (2012), 'Isolation-based anomaly detection', *ACM Transactions on Knowledge Discovery from Data (TKDD)* **6**(1), 3.

Lourenço, H. R., Martin, O. C. & Stützle, T. (2003), Iterated local search, *in* 'Handbook of metaheuristics', Springer, pp. 320–353.

Macha, M. & Akoglu, L. (2018), 'Explaining anomalies in groups with characterizing subspace rules', *Data Mining and Knowledge Discovery* **32**(5), 1444–1480.

Manevitz, L. M. & Yousef, M. (2001), 'One-class svms for document classification', *Journal of Machine Learning Research* **2**(Dec), 139–154.

Matteoli, S., Diani, M. & Corsini, G. (2010), 'A tutorial overview of anomaly detection in hyperspectral images', *IEEE Aerospace and Electronic Systems Magazine* **25**(7), 5–28.

Micenková, B., Dang, X.-H., Assent, I. & Ng, R. T. (2013), Explaining outliers by subspace separability, *in* 'Data Mining (ICDM), 2013 IEEE 13th International Conference on', IEEE, pp. 518–527.

Miller, Z., Dickinson, B., Deitrick, W., Hu, W. & Wang, A. H. (2014), 'Twitter spammer detection using data stream clustering', *Information Sciences* **260**, 64–73.

Mladenović, N. & Hansen, P. (1997), 'Variable neighborhood search', *Computers & operations research* **24**(11), 1097–1100.

Nakariyakul, S. & Casasent, D. P. (2009), 'An improvement on floating search algorithms for feature subset selection', *Pattern Recognition* **42**(9), 1932–1940.

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

Novaković, J. (2016), 'Toward optimal feature selection using ranking methods and classification algorithms', *Yugoslav Journal of Operations Research* **21**(1).

Peng, H., Long, F. & Ding, C. (2005), 'Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy', *IEEE Transactions on Pattern Analysis & Machine Intelligence* (8), 1226–1238.

Pevnỳ, T. & Kopp, M. (2014), Explaining anomalies with sapling random forests, *in* 'Information Technologies-Applications and Theory Workshops, Posters, and Tutorials (ITAT 2014)'.

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

Polat, K., Sahan, S., Kodaz, H. & Günes, S. (2005), A new classification method for breast cancer diagnosis: feature selection artificial immune recognition system (fs-airs), *in* 'International Conference on Natural Computation', Springer, pp. 830–838.

Portnoy, L., Eskin, E. & Stolfo, S. (2001), Intrusion detection with unlabeled data using clustering, *in* 'In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001', pp. 5–8.

Quinlan, J. R. (1986), 'Induction of decision trees', *Machine learning* **1**(1), 81–106.

Radovic, M., Ghalwash, M., Filipovic, N. & Obradovic, Z. (2017), 'Minimum redundancy maximum relevance feature selection approach for temporal gene expression data', *BMC bioinformatics* **18**(1), 9.

Rajasegarar, S., Leckie, C., Palaniswami, M. & Bezdek, J. C. (2006), Distributed anomaly detection in wireless sensor networks, *in* 'Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on', IEEE, pp. 1–5.

Resende, M. G. (2009), 'Greedy randomized adaptive search procedures: Greedy randomized adaptive search procedures gpasp', *Encyclopedia of optimization* pp. 1460–1469.

Saeys, Y., Inza, I. & Larrañaga, P. (2007), 'A review of feature selection techniques in bioinformatics', *bioinformatics* **23**(19), 2507–2517.

Sari, A. (2015), 'A review of anomaly detection systems in cloud networks and survey of cloud security measures in cloud storage applications', *Journal of Information Security* **6**(02), 142.

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

Scott, D. W. (2010), 'Scott's rule', *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(4), 497–502.

Shi, Y. et al. (2001), Particle swarm optimization: developments, applications and resources, *in* 'Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)', Vol. 1, IEEE, pp. 81–86.

Siddiqui, M. A., Fern, A., Dietterich, T. G. & Wong, W.-K. (2015), 'Sequential feature explanations for anomaly detection', *arXiv preprint arXiv:1503.00038* .

Siddiqui, M. A., Fern, A., Dietterich, T. G. & Wong, W.-K. (2019), 'Sequential feature explanations for anomaly detection', *ACM Trans. Knowl. Discov. Data* **13**(1), 1:1–1:22.
URL: *http://doi.acm.org/10.1145/3230666*

Silverman, B. W. (1986), *Density estimation for statistics and data analysis*, Vol. 26, CRC press.

Somol, P., Pudil, P., Novovičová, J. & Paclık, P. (1999), 'Adaptive floating search methods in feature selection', *Pattern recognition letters* **20**(11-13), 1157–1163.

Tarabalka, Y., Haavardsholm, T. V., Kåsen, I. & Skauli, T. (2009), 'Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and gpu processing', *Journal of Real-Time Image Processing* **4**(3), 287–300.

Tax, D. M. & Duin, R. P. (2004), 'Support vector data description', *Machine learning* **54**(1), 45–66.

Van Laarhoven, P. J. & Aarts, E. H. (1987), Simulated annealing, *in* 'Simulated annealing: Theory and applications', Springer, pp. 7–15.

Vinh, N. X., Chan, J., Bailey, J., Leckie, C., Ramamohanarao, K. & Pei, J. (2015), Scalable outlying-inlying aspects discovery via feature ranking, *in* 'Advances in Knowledge Discovery and Data Mining', Springer, pp. 422–434.

Vinh, N. X., Chan, J., Romano, S., Bailey, J., Leckie, C., Ramamohanarao, K. & Pei, J. (2016), 'Discovering outlying aspects in large datasets', *Data Mining and Knowledge Discovery* pp. 1–36.

Voudouris, C. & Tsang, E. P. (2003), Guided local search, *in* 'Handbook of metaheuristics', Springer, pp. 185–218.

Wallace, B. C. & Dahabreh, I. J. (2012), Class probability estimates are unreliable for imbalanced data (and how to fix them), *in* 'Data Mining (ICDM), 2012 IEEE 12th International Conference on', IEEE, pp. 695–704.

Wallace, B. C. & Dahabreh, I. J. (2014), 'Improving class probability estimates for imbalanced data', *Knowledge and information systems* **41**(1), 33–52.

Whitley, D. (1994), 'A genetic algorithm tutorial', *Statistics and computing* **4**(2), 65–85.

Williams, G., Baxter, R., He, H., Hawkins, S. & Gu, L. (2002), A comparative study of rnn for outlier detection in data mining, *in* 'Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on', IEEE, pp. 709–712.

Wong, W.-K., Moore, A., Cooper, G. & Wagner, M. (2002), Rule-based anomaly pattern detection for detecting disease outbreaks, *in* 'AAAI/IAAI', pp. 217–223.

Xue, F., Yan, W., Roddy, N. & Varma, A. (2006), Operational data based anomaly detection for locomotive diagnostics., *in* 'MLMTA', pp. 236–241.

Zhang, B., Sconyers, C., Byington, C., Patrick, R., Orchard, M. & Vachtsevanos, G. (2008), Anomaly detection: A robust approach to detection of unanticipated faults, *in* 'Prognostics and Health Management, 2008. PHM 2008. International Conference on', IEEE, pp. 1–8.

Zhang, J., Lou, M., Ling, T. W. & Wang, H. (2004), Hos-miner: a system for detecting outlyting subspaces of high-dimensional data, *in* 'Proceedings of the Thirtieth international conference on Very large data bases-Volume 30', VLDB Endowment, pp. 1265–1268.

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