Interpreting Data Anomalies: From Descriptive to Predictive Anomaly Explanations

Nikolaos Myrtakis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Prof. *Vassilis Christophides*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The research work was partially supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: 1941). The work has been additionally supported by SAP France.

University of Crete
Computer Science Department

Interpreting Data Anomalies: From Descriptive to Predictive Explanations

Thesis submitted by
Nikolaos Myrtakis
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: ______________________________
Nikolaos Myrtakis

Committee approvals: ______________________________
Vassilis Christophides
Professor, Thesis Supervisor, University of Crete, Greece

____
Ioannis Tsamardinos
Professor, Committee Member, University of Crete, Greece

____
Themis Palpanas
Professor, Committee Member, University of Paris, France

Departmental approval: ______________________________
Polyvios Pratikakis
Assistant Professor, Director of Graduate Studies

Heraklion, September 2020

Interpreting Data Anomalies: From Descriptive to Predictive Explanations

Abstract

In many data exploratory tasks, abnormal and rarely occurring patterns called anomalies (outliers, novelties) are more interesting than the prevalent ones. For instance, they could represent systematic errors, frauds in bank transactions, intrusions in network and system monitoring or other interesting phenomena. Numerous algorithms have been proposed for detecting anomalies. Unfortunately, unsupervised detectors in general, do not explain why a given sample (record) was labelled as an anomaly and thus diagnose its root causes.

Anomaly explanations often take the form of feature subsets of significantly lower dimensionality compared to the original feature space. By examining only the features of an *explaining subspace* suffices to determine whether a sample is an anomaly or not according to a detector. Explanations can be categorized as (i) *descriptive* in the sense that they explain the samples used to train the detector and (ii) *predictive* that generalize to unseen data. In this thesis we experimentally evaluate the main descriptive explanation methods proposed in the literature, as well as, introduce the first predictive explanation method that is inspired by recent advances in Automated Machine Learning systems (AutoML).

In the first part of our thesis, we present a thorough evaluation framework of unsupervised explanation algorithms for individual and groups of anomalies aiming to uncover several missing insights from the literature such as: (a) Is it effective to combine any explanation algorithm with any off-the-shelf outlier detector? (b) How is the behavior of an outlier detection and explanation pipeline affected by the number or the correlation of features in a dataset? and (c) What is the quality of summaries in the presence of outliers explained by subspaces of different dimensionality? A major drawback of the descriptive explanation methods stems from the fact that they should be recomputed for every new batch of data.

To address this limitation, in the second part of our thesis, we present the design and experimental evaluation of the **PROTEUS** AutoML pipeline. PROTEUS produces global, predictive explanations using a surrogate model, specifically designed for feature selection on imbalanced datasets in order to best approximate the decision surface of any unsupervised detector. Computational experiments confirm the efficacy and robustness of PROTEUS to produce predictive explanations for different families of anomaly detectors as well as its reliability to estimate their predictive performance in unseen data.

Ερμηνεύοντας Ανωμαλίες σε Δεδομένα: Από Περιγραφικές σε Προβλεπτικές Εξηγήσεις

Περίληψη

Σε πολλές εργασίες διερεύνησης δεδομένων, ακανόνιστα ή σπανίως εμφανιζούμενα μοτίβα που ονομάζονται ανωμαλίες (αποκλίνοντα ή πολύ διαφορετικά δεδομένα), είναι συχνά πιο ενδιαφέροντα από τα συνήθη μοτίβα. Για παράδειγμα, ακανόνιστα μοτίβα μπορεί να αναπαριστούν συστηματικά σφάλματα, απάτες σε τραπεζικές συναλλαγές, παρεισφρήσεις δικτύων και συστημάτων ελέγχου ή άλλα ενδιαφέροντα φαινόμενα. Πολυάριθμοι αλγόριθμοι έχουν προταθεί για την ανίχνευση ανωμαλιών. Δυστυχώς, οι περισσότεροι ανιχνευτές χωρίς επίβλεψη δεν προσφέρουν κάποια εξήγηση σχετικά με το γιατί ένα δοσμένο δείγμα (καταγραφή) χαρακτηρίστηκε σαν ανωμαλία και ως εκ τούτου να διαγνωστούν οι αιτίες που προκλήθηκε.

Οι εξηγήσεις ανωμαλιών συχνά παίρνουν τη μορφή υποσυνόλων γνωρισμάτων, σημαντικά μειωμένης διάστασης σε σύγγριση με τον αρχικό χώρο γνωρισμάτων. Εξετάζοντας μόνο τα γνωρίσματα σε έναν επεξηγηματικό υπόχωρο, αρκεί ώστε να καθοριστεί εάν ένα δείγμα είναι ανωμαλία ή όχι σύμφωνα με έναν ανιχνευτή. Οι εξηγήσεις μπορούν να κατηγοριοποιηθούν στις εξής (ι) περιγραφικές με την έννοια ότι εξηγούν μόνο τα δείγματα που εκπαιδεύτηκε ο ανιχνευτής και (ιι) περιγραφικές οι οποίες γενικεύονται και σε απαρατήρητα δεδομένα. Σε αυτήν την εργάσία, αποτιμούμε πειραματικά τους κύριες περιγραφικές μεθόδους εξήγησης που έχουν προταθεί στην βιβλιογραφία, καθώς επίσης εισάγουμε την πρώτη μέθοδο για προβλεπτική εξήγηση, εμπνευσμένη από πρόσφατες εξελίξεις στο πεδίο της Αυτοματοποιημένης Μηχανικής Μάθησης (ΑυτοΜΛ).

Στο πρώτο κομμάτι αυτής της εργασίας, παρουσιάζουμε ένα διεξοδικό πλαίσιο αποτίμησης αλγορίθμων εξήγησης ανωμαλιών χωρίς επίβλεψη, τόσο για μεμονομένες όσο και για ομάδες ανωμαλιών με στόχο την αποσαφήνιση διαφόρων αναπάντητων ερωτημάτων από την τρέχουσα βιβλιογραφία όπως: (α) Πόσο εποτελεσματικός είναι ο συνδιασμός οποιουδήποτε αλγόριθμου εξήγησης με έναν οποιονδήποτε ανιχνευτή; (β) Πώς επιρρεάζεται η συμπεριφορά μιας αλληλουχίας ανίχνευσης και εξήγησης ανωμαλιών από τον αριθμό ή την συσχέτιση των γνωρισμάτων στα δεδομένα; (γ) Ποια είναι η ποιότητα μιας σύνοψης στην περίπτωση που οι ανωμαλίες εξηγούνται από υποχώρους διαφορετικών διαστάσεων; Ένα μεγάλο ελάττωμα των περιγραφικών μεθόδων εξήγησης, πηγάζει από το γεγονός ότι πρέπει να ξανα υπολογιστούν για κάθε νέα παρτίδα δεδομένων.

Για να καταπολεμήσουμε αυτόν τον περιορισμό, στο δεύτερο κομμάτι αυτής της εργασίας, παρουσιάζουμε τη σχεδίαση και την πειραματική αποτίμηση του ΠΡΟΤΕΥΣ (Πρωτέας), ενός συστήματος αυτοματοποιημένης μηχανικής μάθησης. Ο ΠΡΟΤΕΥΣ παράγει καθολικές, προβλεπτικές εξηγήσεις χρησιμοποιώντας ένα υποκατάστατο μοντέλο, ειδικά σχεδιασμένο για επιλογή γνωρισμάτων σε μη ισορροπημένα δεδομένα

ώστε να προσσεγγίσει με τον καλύτερο δυνατό τρόπο την επιφάνεια επιλογής οποιουδήποτε ανιχνευτή χωρίς επίβλεψη. Υπολογιστικά πειράματα επιβεβαιώνουν την αποτελεσματικότητα και συνέπεια του ΠΡΟΤΕΥΣ στην παραγωγή προβλεπτικών εξηγήσεων για διαφορετικές οικογένειες ανιχνευτών ανωμαλιών καθώς και την αξιοπιστία του στην εκτίμηση της προβλεπτικής επίδοσης σε απαρατήρητα δεδομένα.

Ευχαριστίες

Θα ήθελα πρωτίστως να ευχαριστήσω τον επόπτη καθηγητή μου κύριο Βασίλη Χριστοφίδη για τη στήριξη, την υπομονή και τα ερευνητικά ερεθίσματα που μου μετέδωσε κατά τη διάρκεια μιας ιδιαιτέρως προσοδοφόρας συνεργασίας. Το σημαντικότερο μάθημα του κύριου Χριστοφίδη αποτέλεσε το παράδειγμα της επιμονής για την αναζήτηση της αιτίας πίσω από τις πειραματικές αποτιμήσεις. Όπως μου έλεγε συχνά *«Τα πειράματα δεν θα σου πουν τίποτα που δεν τα ρωτάς»*. Θα ήθελα να ευχαριστήσω τον καθηγητή κύριο Ιωάννη Τσαμαρδίνο για την συνεπίβλεψη του στην δουλειά του δευτέρου μέρους της εργασίας, ο οποίος εκτός από τις ουσιαστικές συμβουλές και προτάσεις, μου μετέδωσε τον ενθουσιασμό για την αναζήτηση μιας διαφορετικής προσέγγισης ενός θέματος κοιτάζοντάς το στη ρίζα του. Δεν θα μπορούσα να μην είμαι ευγνώμων για τον επικεφαλή επιστήμονα Eric Simon, που με εμπιστεύτηκε και μου έδωσε την ευκαιρία να ταξιδέψω και να ζήσω για πέντε μήνες στο Παρίσι της Γαλλίας, κάνοντας πρακτική άσκηση σε μια από τις μεγαλύτερες εταιρίες παγκοσμίως, την SAP . Οι εμπειρίες, τα διδάγματα και οι προκλήσεις σε αυτό το ταξίδι λειτούργησαν ως πυξίδα για την ολοκλήρωση αυτής της εργασίας. Θα ήθελα επίσης να ευχαριστήσω τον ακαδημαϊκό μου σύμβουλο κύριο Ιωάννη Τζίτζικα, για όλες τις παραγωγικές συζητήσεις και τις συμβουλές του. Ευχαριστώ τον καθηγητή κύριο Θέμη Παλπάνα που συμμετείχε στην επιτροπή αξιολόγησης της μεταπτυχιακής μου εργασίας καθώς και τα μέλη του ερευνητικού εργαστηρίου mens X machina για όλες τις συζητήσεις και την αλληλεπίδραση που είχαμε. Τέλος, θα ήθελα να επισημάνω ότι αυτή η δουλειά δεν θα είχε φτάσει σε αυτό το σημείο χωρίς την αμέτρητη αγάπη, στήριξη και κατανόηση της αγαπημένης μου Φωτεινής, της αδελφής μου Εμμανουέλας, των γονέων μου Ευαγγελίας και Ευάγγελου, της θείας μου Σοφίας και των παππούδων μου Νίκου και Δέσποινας.

στους γονείς μου

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Detection of “anomalous” samples (records, instances), called *anomaly detection*, is an important problem in machine learning. Detecting and diagnosing data anomalies [1] are important tasks in data processing pipelines used to build industrial-strength Machine Learning (ML) systems [66]. In scientific and industrial monitoring applications, anomaly detection is often the ultimate goal of data analysis as it enables the identification of anomalous samples that may indicate mislabelled data, catastrophic measurements or data entry errors, bugs in data wrangling and preprocessing software, a system that is under attack, about to fail, or other interesting phenomena that decrease the accuracy of the predictive models constructed downstream [63, 91].

Numerous **unsupervised** algorithms of various outlyingness criteria such as IF (isolation-based) [49], LOF (density-based) [10], LODA (projection-based) [65], ABOD (angle-based) [45]) to detect anomalies (hereafter **detectors**) have been proposed. The most advanced ones detect anomalies in a multi-dimensional fashion, simultaneously considering all feature values to call an anomaly. Unfortunately, detectors, in general, do not explain why a sample was considered as abnormal, leaving human analysts with no guidance about their root causes [2], insight to take corrective actions, or remedy their effect [52] (e.g. by repairing data errors or retraining the predictive models for concept drifts).

Several methods for **explaining anomalies** have been proposed, hereafter **explainers**. *The explanations often take the form of a subset of features* called a **subspace** in the literature. The idea is that *by examining only the explaining features suffices to determine whether the sample is an anomaly or not according to the detector*. To illustrate, assume that we have a three dimensional dataset with features F_1, F_2 and F_3 and that we would like to explain the identified anomalous points $o1$ and $o2$ depicted by a black circle and a black square in Figure 1.1-a). In the full feature space of the dataset, $o1$ exhibits a small deviation from most of the other points in the dataset while $o2$ looks like an inlier (normal point) although it exhibits a significant outlyingness when considering the subspace $\{F_2, F_3\}$ (see

[1] In this work we refer to anomalies and outliers interchangeably

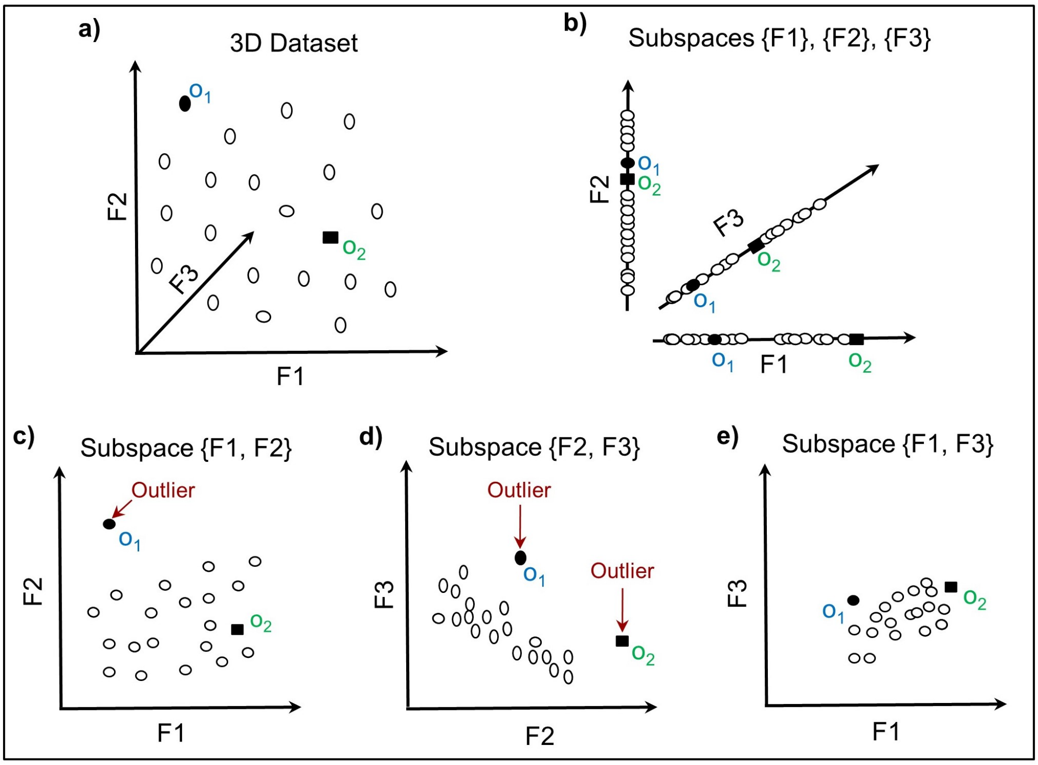


Figure 1.1: A $3d$ dataset with three $1d$ and $2d$ subspaces

Figure 1.1-d). We refer to the former case as *full space* outliers and to the latter as *subspace* outliers. In both cases, we are interested in explaining under which subspaces, called *explaining subspaces*, the given points exhibit high outlyingness. None of the $1d$ subspaces $\{F_1\}$, $\{F_2\}$ and $\{F_3\}$ explain the outlyingness of the two points (see Figure 1.1-b). The same is true for the $2d$ subspace $\{F_1, F_3\}$ (see Figure 1.1-e). Subspace $\{F_1, F_2\}$ explains the outlyingness of $o1$ only (see Figure 1.1-c), while $\{F_2, F_3\}$ explains the outlyingness of both points (see Figure 1.1-d). We can observe that outlyingness of $o1$ is higher in $\{F_1, F_2\}$ than in $\{F_2, F_3\}$. Features contained into the explanation of an outlier are called *relevant*. For instance, F_1 and F_2 are relevant to the explanation of $o2$.

Existing methods can be categorized to those that provide **local explanations** (point-based) that pertain to a single sample, or **global explanations** (a.k.a. explanation summarization) to simultaneously explain all training samples. The latter are important in order to reduce the burden of human analysts having to have to inspect possibly different explanations for each anomaly. An example of a local explanation for point $o1$ is the subspace $\{F1, F2\}$ in Figure 1.1-c), while $\{F2, F3\}$ is a global explanation (see Figure 1.1-d)) . Explainers may be **specific** to a detection algorithm or detector-**agnostic**, hence applicable post-hoc to any detection algorithm. As reported by several independent experimental studies

[27, 18, 11], there is no detector outperforming all others on all possible datasets. Hence, researchers cannot just design a specific explainer for the optimal detector; it may thus be preferable to design optimal agnostic explainers. Explainers may also be categorized as **descriptive** in the sense that they explain the samples used to train the detector. Explainers that return explanations that generalize to unseen data are **predictive** ones. The importance of predictive explanations has been recognised in Explainable AI [57] to avoid recomputing explanations on every new batch of data.

In the first part of our thesis, we evaluate two *point explanation algorithms*, *RefOut* [38] and *Beam* [62], that rank subspaces best explaining the outlyingness of *individual* data points, and two *explanation summarization algorithms*, *LookOut* [29] and *HICS* [36], that rank subspaces best explaining the outlyingness of the majority of the outlier points. Although there exist several efforts for benchmarking outlier detectors in batch [18, 11, 25, 81] and stream [82, 48, 23] processing settings, outlier explanation and summarization algorithms have not yet been thoroughly evaluated under realistic assumptions. To the best of our knowledge, this is the first comprehensive and detailed evaluation of existing algorithms aiming to uncover several insights missing from the existing literature. More precisely, our evaluation yields the following major findings:

1. *Is it effective to combine any explanation algorithm with any off-the-shelf outlier detector?* The majority of explanation mechanisms rely on existing detectors to assess the outlyingness of points in specific feature subspaces. A critical factor of their effectiveness not yet thoroughly evaluated, is how well detectors score outliers in projections or augmentations of the subspaces considered by the search strategy of explainers. A detector is usually expected to highly score outliers when its outlyingness criterion is better suited to the underlying data distribution (e.g., LOF for density-based outliers). Surprisingly enough ABOD (angle-based detector) proved to be more effective than LOF for density-based subspace outliers when used with (i) LookOut in datasets highly contaminated with outliers and (ii) Beam in high dimensional explanations.

2. *How is the behavior of an outlier detection and explanation pipeline affected by the number of features or their correlation in a dataset?* Algorithms achieve *different tradeoffs* between efficiency and effectiveness depending on (i) the dimensionality of explanations, and (ii) the ratio of features in the dataset which are relevant to the outliers. For *full space outliers* (100% feature relevance ratio), *state-wise search* employed by Beam and *exhaustive search* implemented by LookOut achieve the best tradeoff. However, their efficiency significantly decreases in high dimensional datasets. For *subspace outliers*, *random subspace projection* employed by RefOut provides a good tradeoff for a medium ratio of relevant features (35% and 21%) while state-wise search of Beam is the only effective solution for high dimensional explanations ($3d$-$4d$) and datasets (i.e., $<$ 12% ratio of relevant features). When outliers are hidden in subspaces with correlated features in high dimensional datasets, HiCS seems to be the only viable option.

3. *What is the quality of summaries in the presence of outliers explained *by subspaces of different dimensionality?* As the objective of summarization is to discover subspaces where the majority of the outliers seem to deviate w.r.t. inliers, we showed that existing algorithms do not provide guarantees regarding (i) the coverage of the points to be explained; (ii) the overlap or the equivalence of subspaces in the explanation summaries. These properties are essential given that the optimal explanation dimensionality of points is not known in advance.

In the second part of our thesis, we focus on the concept of predictive explanations and propose **the first methodology to produce global, predictive explanations** called **PROTEUS**[2]. PROTEUS is additionally *agnostic*, applicable to any detector. Some notable representatives of prior work on agnostic explainers is the CA-Lasso [56] and the SHAP [51]. LODA [65] is an example of a specific explainer. However, all of the aforementioned explainers are both **local and descriptive**. To that end, we developed PROTEUS, an AutoML pipeline specifically designed to produce surrogate models in this context. It contributes the following design choices:
(1) By definition, anomalies are rare, making it difficult to approximate the detector's decision boundary around the positive class. To improve performance, PROTEUS oversamples the rare class, i.e., the anomalies. In contrast to standard oversampling where pseudo-samples *are assumed* to belong to the minority class [32], PROTEUS uses the detector to label the pseudo-samples.
(2) To select features, PROTEUS employs several feature selection algorithms that are suitable for high-dimensional data and small-sample sizes. Importantly, such algorithms deal with removing not only *irrelevant*, but also *redundant features* [84]. This is in contrast to Feature Importance as calculated by Random Forests for example [22].
(3) To produce the best surrogate model, PROTEUS tries multiple combinations of feature selection and binary classification algorithms tuning their hyperparameters, called *configurations*. The best configuration found is employed to produce the final surrogate model. For the moment, a simple grid-search is employed.
(4) To identify the best configuration, PROTEUS needs to accurately estimate the out-of-sample performance of each configuration. To this end, it employs a special variant of Cross-Validation (**CV**), namely a *group-based, stratified, repeated, K-fold CV with Bootstrap Bias Correction (***BBC***)* [87]. This variant addresses the issues of over-sampling, multiple tries of configurations, low sample size, and imbalancing of the anomaly class. Experiments show that it provides better estimates than standard alternatives.

Together, the above design choices guarantee that *PROTEUS will (a) identify a high-performing surrogate model with few features, provided there is one, and (b) will accurately estimate its out-of-sample performance in predicting the anomaly detector's behavior.* The above statements are supported by experiments on several real and synthetic datasets. Unlike ad-hoc or detector-specific feature importance

[2] Proteus or *Πρωτεύς* in Greek, means 'first' and is a minor sea God and son of Poseidon.

methods proposed in the literature, our experiments also demonstrate *PROTEUS robustness to increasing data dimensionality.*

Chapter 2

Unsupervised Anomaly Detectors

Several methods have been proposed in the literature to measure the abnormality of a data point in a dataset. In this thesis, we survey four unsupervised methods that are widely used for detecting outliers in datasets with multiple numerical[1] features [18, 11, 25, 81]. We should also stress that in this work, we did not include algorithms that run exclusively in stream mode [42, 78, 28].

The outlyingness criteria underlying each method have respective strengths and weaknesses w.r.t. the characteristics of the datasets (e.g., dimensionality) and outliers (e.g., highly clustered or not).

Density-Based methods, such as Local Outlier Factor (LOF) [10] take into account the local density of points when searching for outliers. An example of outliers detected by LOF is illustrated in Figure 2.1-a). The point $o1$ is considered to be an outlier as it lies on a sparse area while its nearest neighbors lie on dense areas. The distance of a point p from o is computed using the following *reachability distance* (reach-dist):

$$\text{reach-dist}_k(p \leftarrow o) = max\{k\text{-dist}(o), d(p, o)\}$$

where k-dist(o) is the distance of o to its kth nearest neighbor and $d(p, o)$ is the direct distance (e.g., Euclidean) between the two points. LOF computes the local reachability density of a point p as the inverse of the average reachability distance of p from its k-nearest neighbors (kNN):

$$\text{lrd}_k(p) = 1/(\text{mean}_{o \in kNN(p)} \text{reach-dist}_k(p \leftarrow o))$$

Finally, the density of a point is compared to the average local reachability density of its neighbors to obtain a *score*:

$$\text{LOF}_k(p) = \text{mean}_{o \in kNN(p)} \frac{\text{lrd}_k(o)}{\text{lrd}_k(p)}$$

[1] Anomaly detection methods for categorical data [79] are outside the scope of this work.

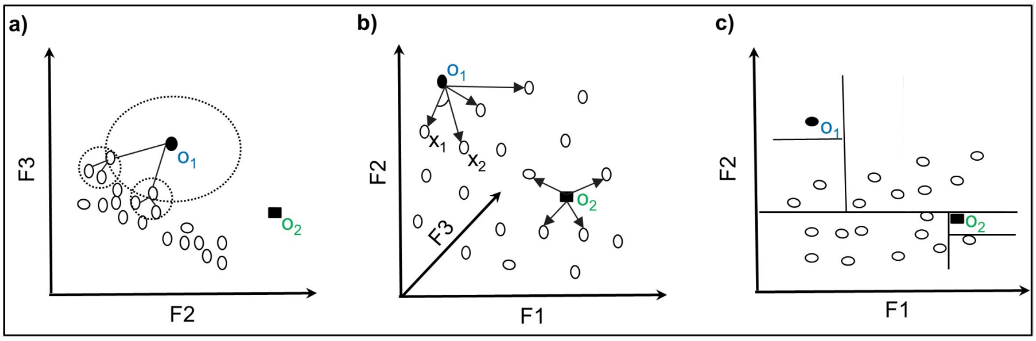


Figure 2.1: Examples of outliers in different subspaces detected by (a) LOF, (b) Fast ABOD and (c) iForest

LOF's time complexity for training is $O(1)$ since there is no training step and $O(n^2)$ for prediction, where n is the number of points in a dataset. Inliers obtain scores around 1 while outliers obtain scores significantly larger than 1. LOF distinguishes effectively outliers from inliers in regions of *varying density* where outliers lie on highly sparse areas far from dense clusters.

Isolation-Based methods estimate the probability of a point to be an outlier on the basis of the number of partitions needed to isolate it from the other points in a dataset. The less partitions needed to isolate, the more likely a data point is to be an outlier. For instance, in Figure 2.1-c) the point o_1 is an outlier as it needs less partitions to be isolated compared to the inlier o_2.

Isolation Forest (iForest) [49] exploits this property using a forest of random trees built on samples of the dataset by uniformly selecting features and their split values. The outlyingness score of a data point is then computed by averaging over all trees the path length from the root to the leaf node with the data point:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

The score assigned to points is normalized within the range [0,1], with anomalies getting a score close to 1. iForest has a small memory-footprint ($O(tn)$), where t is the number of trees and n is the subsample size. It achieves a sublinear time-complexity ($O(tnlogn)$) for training by exploiting subsampling and by eliminating the heavy cost of distance computation and ($O(tlogn)$) for prediction. Being agnostic to the distances (or densities) of points, iForest is able to detect anomalies effectively even if they are lying on less dense areas than the majority of the points.

Angle-Based methods compute for each given point, the angles to other data points N. The Angle Based Outlier Detector (ABOD) [45] uses the variance of these angles as an outlyingness score. For example, as we can see in Figure 2.1-b), $o1$ is an outlier as its neighbors are located in similar directions (small angle variance), but o_2 is an inlier as it is surrounded by its neighbors in various directions (high angle variance). The ABOD score for a given point $o1$ and any pair of points $x1$, $x2$ is computed as:

$$ABOD(o_1) = \underset{x_1,x_2 \in N}{\mathrm{Var}} \left(\frac{\langle \overrightarrow{x_1 o_1}, \overrightarrow{x_2 o_1} \rangle}{\|\overrightarrow{x_1 o_1}\|^2 \cdot \|\overrightarrow{x_2 o_1}\|} \right)$$

ABOD's time complexity for training is $O(1)$ since there is no training step and $O(N^3)$ for prediction. Due to ABOD's high computational cost, we are focusing on the efficient variant of the original algorithm, called *Fast ABOD*, which computes the angles of a particular point only to its k-nearest neighbors. Small angle variance results high ABOD score indicating high outlyingness. Intuitively, a point is more likely to be an outlier when it lies on the borders of the data distribution. ABOD avoids to compute the distance between points, hence it is a suitable detector for high dimensional datasets.

Projection-based histogram ensemble detectors like LODA [65], constructs an ensemble of t one-dimensional histogram density estimators. Each density estimator, p_j, $j = 1, \ldots, t$ is constructed as follows. First, a projection vector w_j is initialized as the d-dimensional zero vector, then $k = \sqrt{d}$ features are selected at random, and finally those positions in w_j are replaced by standard normal random variates. Each training sample x_i is then projected to the real line as $w_j^T x_i$, and a *fixed-width histogram* density estimator p_j is estimated via the method of Birgé and Rozenholc [8]. To compute the anomaly score of a test sample x_i', LODA computes the average log density:

$$f(x_i') = \frac{1}{t} \sum_{j=1}^{t} \ log \ p_j(w_j^T x_i')$$

Averaging the scores of individual histograms over all anomalous test samples can be used to sort features in decreasing order, which means that features in which anomalous samples deviates the most should be the first. A significant advantage of LODA compared to LOF and iForest is that it can handle missing variables. LODA's time complexity given E histograms with b bins, is $O(nt(d^{-\frac{1}{2}} + b))$ for training and $O(t(d^{-\frac{1}{2}} + b))$ for prediction where n is the number of points in the dataset and d its dimensionality.

Chapter 3

Related Work

In this section, we first give the definitions of anomaly detection and explanation and briefly survey various categories of related work on the explanation of anomalies.

Unsupervised Anomaly Detectors. Let $D = \{x_1, \ldots, x_n\}$ be a dataset of n samples, where each sample $x_i \in \mathbb{R}^d$. An **Anomaly Detector** A is a function that accepts a dataset D and produces an Anomaly Detector Model $\hat{Y}_A$. An **Anomaly Detector (Classification) Model** $\hat{Y}_A$ is a function $\mathbb{R}^d \rightarrow \{0,1\}$. The value $\hat{Y}_A(x) = 1$, semantically denotes the identification of an anomalous sample. We note that many anomaly detection algorithms may output models that return "anomalousness" (numerical) scores instead of binary decisions. These models can be converted to classification models by thresholding the scores by some value t. Typically, t is set to the expected ratio of anomalies in the dataset [11]. Statistical techniques have been proposed [92] to fine-tune threshold t.

Definition 1. *The (descriptive) explanation $e_D(a_i)$ of an anomaly $a_i \in D$, is a subset of features $e_D(a_i) = \{f_{ij} \mid j = 1, \ldots, k\}$, where $k \ll d$, that captures most of a_i's anomalousness, i.e.,*

$$\hat{Y}_A(a_i) = \begin{cases} 1, & \text{iff } score(a_i[e_D(a_i)]) > t \\ 0, & \text{iff } score(a_i[e_D(a_i)]) \leq t \end{cases} \tag{3.1}$$

where $[\cdot]$ denotes the projection of a_i over the features $e_D(a_i)$ composing its explanation.

Such explanations are called *descriptive* as they are computed for every set of anomalies and normal points. In order to make explanations also *discriminative* for a new batch of data, we need to transform the unsupervised anomaly detection into a supervised classification problem. To this end, we can train a classification algorithm $c(X, \hat{Y}_A(X))$ where $X \in \mathbb{R}^{n \times d}$ is the multi-dimensional data points and $\hat{Y}_A(X) \in \{0,1\}$ their labels as predicted by detector A. We should stress that the quality of the predictive model strongly depends on the quality of the unsupervised detector to separate anomalies from normal points.

Definition 2. *The (predictive) explanation $e_P(X)$, is a minimal subset of features that leads to an optimal predictive model on the outcome of an anomaly detector w.r.t. a performance metric* Perf, *i.e.,* $\neg\exists e'_p(X) \subseteq e_p(X)$ *such that:*

$$\text{Perf}(c(X[e_p(X)], \hat{Y}_A(X)) \leq \text{Perf}(c(X[e'_p(X)], \hat{Y}_A(X))$$

In chapter 7 we explain and justify the performance metric we used for assessing the quality of a predictive explanation.

We now present various works for explaining data anomalies in unsupervised and supervised settings, partially inspired by [52]. We should stress that explanation of anomalies in temporal data is beyond the scope of this work [24, 7].

3.1 Explaining Outliers in Query Answers

Scorpion [89] was the first system for explaining outliers in the result of group-by queries. Given a set of outliers spotted by analysts on the results of queries, the system searches for a logical formulae that describes a set of tuples that contribute most to the excessively high or low aggregate value of a specific group. It is hard to extend this work for explaining outliers recognized by off-the-self detectors. Furthermore, empirical explanations for data points that violate specific data quality constraints (i.e., inconsistencies w.r.t. domain-specific rules) have been studied in [16]. A glitch explanation is a collection of values of features that have statistically significant propensity signatures. In our work, we are interested in a quantitative form of data anomalies frequently encountered in transaction or measurement-based datasets, i.e., outliers in numerical features for which quality constraints are difficult or impossible to obtain. Finally, an interactive explanation discovery system has been proposed [69]. It relies on a set of explanation templates given by analysts that need to be precomputed in a given dataset. Neither of the previous methods satisfy our requirements for explaining data anomalies in a way that is both domain and detector agnostic without making strong assumptions regarding how the input datasets have been processed.

3.2 Explaining Outliers in Temporal Data

MacroBase [1] enables efficient, accurate, and modular analyses that highlight and aggregate important and unusual behavior in fast data. It introduces an operator for explaining outliers in a data stream based on the categorical features rather than the numerical features used to actually detect outliers. In contrast to the notion of relevant subspaces, the explanation of continuous outliers consists of conjunctions of categorical features whose values cover most of the outliers detected by a density-based method called MAD. ExplainIT [35] is a recent system for unsupervised root-cause analysis of time series that shares similar motivations with MacroBase. It empowers a declarative interface (SQL based) for specifying a large number of cause hypothesis that need to be tested and ranked to assist analysts with a reduced number of causal dependencies that have to exploit regarding an observed phenomenon. The use of causal models for explaining data outlyingness is an interesting idea that we plan to study in the future by leveraging our previous work on scalable algorithms for causal feature discovery [85]. Finally, EXstream [93] is a system providing high-quality explanations for anomalous behaviors of streaming data that analysts annotate using CEP-based monitoring results. Explanations take the form of logical formulae in CNF involving relational predicates (i.e., $=$, $<$, $\leq$) over feature values computed for time series. Authors formalize the problem of optimally explaining anomalies in CEP as an information reward maximization problem. In this respect, an entropy-based distance function of time series is used to measure the contribution in the reward of each feature. As the reward function is sub-modular, greedy approximation techniques could be used as in the case of LookOut [29]. Computing explanations based on single-feature rewards bears similarity with the univariate feature selection problem while computing subspace based outlier explanations is closer to the more complex problem of multivariate feature selection [85].

3.3 Explainable Anomaly Detectors

Given that unsupervised detectors assess the abnormality of multidimensional data on various feature subspaces, they can also report as explanations which subspaces contributed the most to the score of a point. A first example of such explainable outlier detectors is LODA [65] which computes the anomaly score of a data point as the average log density over an ensemble of one-dimensional histogram density estimators. Given that each histogram with sparse projections provide an anomaly score on a randomly generated subspace, LODA explains the scores by ranking the features according to their contribution to point's anomalousness.

LODI [15] and LOGP [14] seek an optimal subspace in which an outlier is maximally separated from its neighbors. Both works perform a dimensionality reduction technique and measure the outlyingness in a low-dimensional subspace capable of preserving the locality around the neighbors while at the same time maximizing the distance from the candidate point. Then, the top-k features with the largest absolute coefficient from the eigenvector with the largest eigenvalue are selected and returned as explanation of a candidate point.

[72] proposes an interactive explanation method that can be instantiated for any anomaly detection scheme based on density estimation. [43] introduced a method to detect outliers in axis-parallel subspaces, called SOD, that computes the anomaly score of a point in a hyperplane w.r.t. to nearest neighbors in the full space. SOD hyperplanes that contribute most in the anomaly scores could be used as explanations. CMI [9] and HiCS [37] rely on statistical methods to select subspaces of high-dimensional datasets, where anomalies exhibit a high deviation from normal points. Both consider highly contrasting subspaces as explanations

of all possible anomalies in a dataset.

The aforementioned works mostly explain anomalies as a byproduct of the unsupervised detection method. Given that independent experimental evaluations showed that no detector outperforms all others for all possible datasets [26, 11, 18, 27], in our work we focus on learning the decision boundary of any unsupervised anomaly detector using the available ground truth. In contrast to the descriptive explanations provided by the aforementioned works, Proteus targets predictive explanations that could be successfully also for unseen data.

3.4 Post-hoc Anomaly Explainers

The primary focus of these methods is to specify a subset of features such that a data point may obtain a high anomaly score when projected onto these subspaces. Some authors have referred to this explanation task as "outlying aspects mining" [20, 61].

The following works perform local explanations aiming to explain individual anomalies. The seminal work [40] first introduced the problem of explaining individual outliers with "Intentional knowledge" under the form of minimal feature subspaces in which they show the greatest deviation from inliers. To find optimal subspaces, [46] formulates a constraint programming problem to maximize differences between neighborhood densities of known outliers and inliers. [39] employs a search strategy aiming to find a subspace which maximizes differences in anomaly score distributions of all points across subspaces while [56] measures the separability between outlier and inliers as the classification accuracy between the two classes, and then apply supervised feature selection methods to produce a local explanation. OAMiner [20] finds the most outlying subspace where the data point is ranked highest in terms of a probability density measure and OARank [61] ranks features based on their potential contribution toward the anomalousness of a data point.

Extending earlier work [3] on explaining individual outliers, [4, 5] focus on explaining groups of anomalies for categorical data using contextual rule based explanations. Authors search for <context, feature> pairs, where the (single) feature can differentiate as many outliers as possible from inliers that share the same context. The outlyingness score for a data point in a subspace is calculated based on the frequency of the value that the outlier takes in the subspace. It tries to find subspaces E and S such that the outlier is frequent in one and much less frequent than expected in the other. To avoid searching exhaustively all such rules, the method takes two parameters, and, to constrain the frequencies of the given data point in subspaces E and S, respectively. Similarly, [94] describes anomalies grouped in time. They construct explanatory Conjunctive Normal Form rules using features with low segmentation entropy, which quantifies how intermixed normal and anomalous points are. They heuristically discard highly correlated features from the rules to get minimal explanations. The previous related work assumes that outliers are scattered and strive to explain them individually rather than to summarize the explanation of a collection of outliers.

The following works perform explanation summarization aiming to explain a set of anomalies collectively rather than individually. LookOut [29] exploits a submodular optimization function to ensure concise summarization. xPACS [52] groups anomalies by generating sequential feature-based explanations providing a ranked list of feature-value pairs that are incrementally revealed until the human expert reaches a satisfactory level of confidence. In contrast to the interactive explanations provided by xPACS, Proteus provides global feature subspaces that could potentially explain even out-of-sample anomalies.

3.5 Visual, Interactive Exploration of Outliers

VSOutlier [12] is a system for supporting interactive exploration of outliers in Big Data streams. It integrates various distance-based continuous detectors of outliers [82, 48] and provides a rich set of interactive interfaces to explore outliers in real time. While such visualizations may provide a simple context to understand outliers, they do not offer clues to explain abnormality in high-dimensional data. Human analysts are not assisted to choose $2d$ plots that are actually projections of high dimensionality subspaces relevant to the outliers of interest. A framework of sequential feature explanations (SFEs) of data anomalies has been presented in [73]. An SFE of an anomaly is a sequence of features, which are presented to the analyst in order until the information contained in the highlighted features is enough for the analyst to make a confident judgement about the anomaly. Authors formalize the problem of optimizing SFEs for a particular density-based outlier detector and present both greedy algorithms and an optimal algorithm (based on branch-and-bound search) for SFEs. Rather than presenting to analysts interactively the features of a relevant subspace per given outlier, we are focusing in this work on algorithms summarizing explanations of a set of given points to reduce the burden of analysts.

3.6 Explaining Black-box Predictors

Several methods have been recently proposed to explain why a supervised model predicted a particular label for a particular example [21, 41, 58, 67]. LIME [67] constructs a linear interpretable model that is locally faithful to the predictor. In this respect, it draws uniformly at random (where the number of such draws is also uniformly sampled) pseudo-samples per every point to be explained. Note that LIME let the black-box classifier label the generated pseudo-samples. To the best of our knowledge, LIME has not been successfully used for imbalanced neighborhoods [88]. Other works [21, 41] explain the model by perturbing the features to quantify their influence on predictions. However, these works do not aim to explain multiple examples collectively, as the global explanation problem studied in our work.

Other works aim to produce explanations in the form of feature relevance scores, which indicate the relative importance of each feature to the classification decision. Such scores have been computed by comparing the difference between a classifier's prediction score and the score when a feature is assumed to be unobserved [68], or by considering the local gradient of the classifier's prediction score with respect to the features for a particular example [6].

[75, 76] considered how to score features in a way that takes into account the joint influence of feature subsets on the classification score, which usually requires approximations due to the exponential number of such subsets.

The aforementioned works require as input a supervised model rather than an unsupervised anomaly detector. However, in real application settings it is difficult or even impossible to label data as anomalous or normal examples [26]. Moreover, Proteus provides global explanations returned by standard feature selection algorithms after learning the decision boundary of the unsupervised detector.

3.7 Evaluation of Explainers

Existing approaches for evaluating explanation methods in both supervised and unsupervised settings are typically quite limited in their scope. Often evaluations are limited to visualizations or illustrations of several example explanations [6, 14] or to test whether a computed explanation collectively conforms to some known concept in the dataset [6], often for synthetically generated data. [72] proposes a larger scale quantitative evaluation methodology for anomaly explanations regarding sequential feature explanation methods. Compared to this study, in our work we assess the predictive performance of a classifier given an explanation along with the correctness of the learned features of the explanation.

3.8 Imbalanced Learning

One of the main challenges in supervised anomaly detection, is class imbalance: anomalies are largely underrepresented compared to normal examples. In the following we position Proteus w.r.t. the main imbalanced learning methods [32]. The imbalanced learning problem is concerned with the performance of learning algorithms in the presence of underrepresented data and severe class distribution skews. We follow the same categorization of imbalanced learning methods as in [32].

Random oversampling augments the original dataset by replicating examples from the minority class, while *random undersampling* removes a random set of majority class examples. Random oversampling may lead to overfitting [55] while undersampling may eliminate useful examples leading to a worse performance. Proteus pipelines do not perform random under/over-sampling. The synthetic minority oversampling technique (SMOTE) [13] generates new minority class examples from the line segments that join the k minority-class nearest neighbors. Our pipeline generates synthetic examples close to the original minority examples by adding gaussian noise. SVM SMOTE [60] is a SMOTE variant that generates the synthetic examples concentrated in the most critical area, i.e., the boundary discovered by fitting an SVM classifier. Borderline-SMOTE [30] seeks to oversample the minority class instances in the borderline areas, by defining a set of "Danger" examples. Adaptive Synthetic Sampling (ADASYN) [31] algorithm uses a density distribution as a criterion to automatically decide the number of synthetic examples that need to be generated for each minority example. In comparison to the aforementioned works, Proteus performs a supervised synthetic minority oversampling ensuring that new data points are anomalies according to the decision boundary of an unsupervised detector that is currently explained. In addition, we proposed a method to avoid information leakage in the cross validation protocol when synthetic oversampling is applied.

Part I

A Comparative Evaluation of Descriptive Explanation Algorithms

Chapter 4

Unsupervised Anomaly Explainers

We are primarily interested in *unsupervised* algorithms that are both *domain-agnostic* (i.e., suitable for datasets from various domains) and *detector-agnostic* (i.e., they can be employed to explain outliers produced by any off-the-self detector). We did not include [44, 70] in our testbed as the explanation is a byproduct of the detection process and thus they do not fulfil the detector-agnostic requirement.

4.1 Point Explanation Algorithms

The objective of a point explanation algorithm is to discover the subspaces that best explain the outlyingness of a multi-dimensional point, i.e. the feature sets where this point deviates most in the dataset. Such subspaces are called *relevant* w.r.t. to the explanation of an outlier. Point explanation algorithms essentially rely on a *search strategy* for exploring feature subspaces in a dataset and an *outlyingness criterion.* The main challenge is that no interesting monotonic property holds for most outlyingness criteria [62], which prevents us to effectively prune the exponential space of feature sets (2^d) w.r.t. data dimensionality (d). Using the detectors presented previously, an outlier discovered in low-dimensional subspaces may become invisible, i.e., masked by inliers in high-dimensional subspaces and vice versa.

RefOut [38] is a sampling based algorithm which employs a stage-wise technique exploiting *random subspace projections* to find relevant subspaces of a *fixed dimensionality.* The main algorithmic steps of RefOut are illustrated in Figure 4.1. Initially, RefOut builds a random pool of size n with random subspace projections drawn from the full feature space of the dataset. In the example of Figure 4.1, we depict a pool of size 4 that contains $3d$ random subspaces (i.e, 50% of the $6d$ dataset). Using an off-the-self detector, the to-be-explained outlier $p1$ is scored in each subspace of the pool. To avoid dimensionality bias when scoring subspaces, the score of a point p in a subspace s, denoted as $score(p_s)$ is standardized using

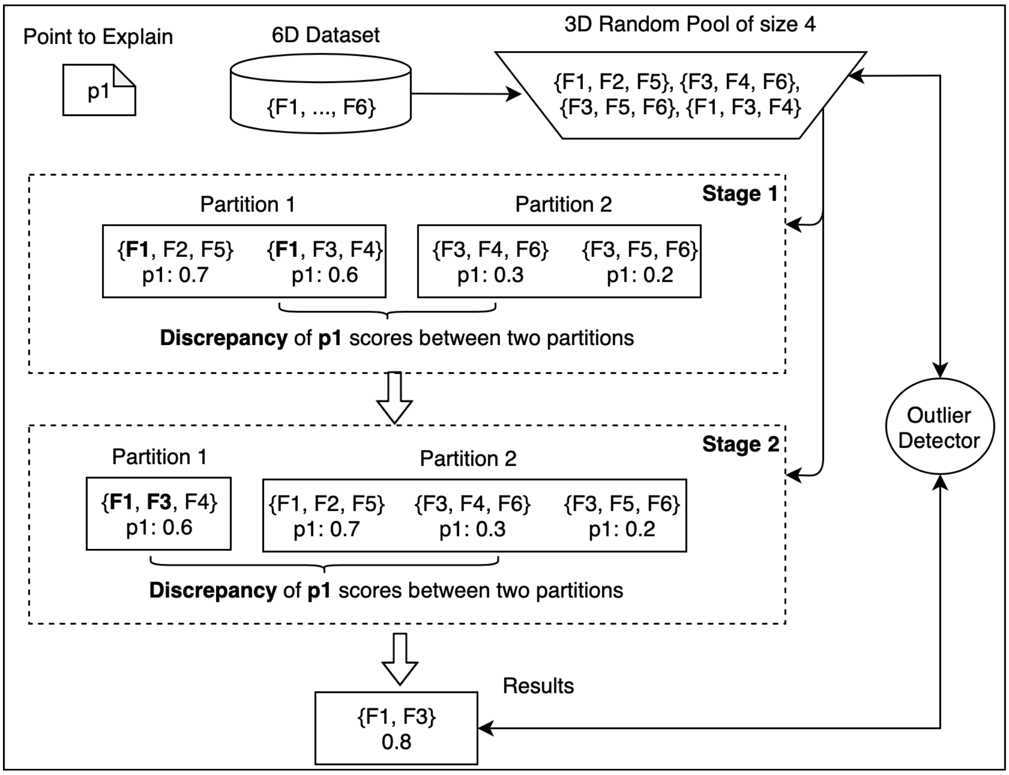


Figure 4.1: RefOut steps to find $2d$ subspaces from a $6d$ dataset to explain the point p1

Z-score as follows:

$$score(p_s)' = \frac{score(p_s) - \overline{score_s}}{\sqrt{Var(score_s)}}$$

RefOut follows a stage-wise technique. In stage 1, RefOut assesses every single feature in the pool. In other words, in this stage it collects the best univariate subspaces. In our example, for the feature $F1$ RefOut partitions the pool into two populations of random subspaces w.r.t. whether they contain or not the feature $F1$. To assess the importance of a feature for explaining the outlyingness of the point $p1$, RefOut quantifies the discrepancy of score populations between the two partitions under the hypothesis that they have equal means. To test this hypothesis, the two-sample Welch's t-test[1] is employed as the two samples may have unequal variances and/or unequal sample sizes. The partitioning is repeated for every feature in the pool and the top-k ones with the highest discrepancy are kept; in our example we kept only $\{F1\}$ for simplicity. In stage 2, RefOut applies the same partitioning and scoring process for $2d$ subspaces by taking the Cartesian product of the top-k subspaces from the previous stage with all the

[1] https://en.wikipedia.org/wiki/Welch's_t-test

univariate subspaces drawn from the pool. In our example, since we are interested in $2d$ explanations the process stops at stage 2 and the best subspace ($\{F1, F3\}$) is returned as explanation of point $p1$. When multiple outliers have to be explained, RefOut searches for relevant subspaces for every point individually.

To sum up, the core idea of RefOut is to make subspace selection adaptive to the outlyingness score of each point and flexible w.r.t. different detectors. It relies on a pool of random subspace projections to assess the important features, that may contribute to the detection of relevant subspaces for a specific point. As feature importance is measured via the discrepancy of outlyingness score distributions, RefOut's effectiveness depends strongly on the ability of an off-the-self outlier detector to assign high scores to outliers. In particular, RefOut makes the assumption that outliers explained in low-dimensional subspaces exhibit a significant outlyingness also in their high-dimensional supersets.

Beam [62] is a *stage-wise* greedy algorithm that takes as input a particular point and returns the subspaces, up to a given dimensionality, that best explain its outlyingness. Although the maximum dimensionality of subspaces returned by Beam is predefined, the algorithm may output subspaces of varying dimensionality. Beam maintains two lists: (i) a *global list* of the best subspaces considered as relevant across stages, (ii) a *stage list* with the best subspaces in each stage. The main algorithmic steps of Beam are illustrated in Figure 4.2 via an example requesting to explain the outlyingness of a point $p1$ of a $6d$ dataset with up to $3d$ subspaces. Using an outlier detector, Beam scores exhaustively in stage 1 all the 15 $2d$ subspaces drawn from the 6 features space of the dataset for the point $p1$. Then, the top-k scored $2d$ subspaces will be inserted both into the *stage list* and *global list*. In stage 2, the best $2d$ subspaces kept in *stage list* will be combined with other features to form $3d$ subspaces as depicted in Figure 4.2. The top-k $3d$ subspaces are then kept in the *stage list*, while the *global list* is updated with the $3d$ subspaces with higher scores for $p1$ than the $2d$ subspaces previously computed. As we required $3d$ explanations in our example, the process will stop at stage 2. The *global list* is then returned as the result of the algorithm.

In a nutshell, Beam is a *stage-wise* greedy algorithm that exploits the top-k best relevant subspaces returned by early stages to search for relevant subspaces in latter stages. Hence, its effectiveness depends strongly on whether a given point obtains a high outlyingness score in *lower projections* of the relevant subspace(s) that should be finally returned. In order to make a fair comparison with RefOut, we report only the best subspaces from the *stage list* in the final stage i.e., subspaces of predefined maximum dimensionality. We call this variation $Beam_{FX}$.

4.2 Explanation Summarization Algorithms

The objective of an explanation summarization algorithm is to discover for a set of outlier points, the subspaces that distinguish as many outliers from inliers as possible. Explanation summarization algorithms also rely on a *search strategy* to

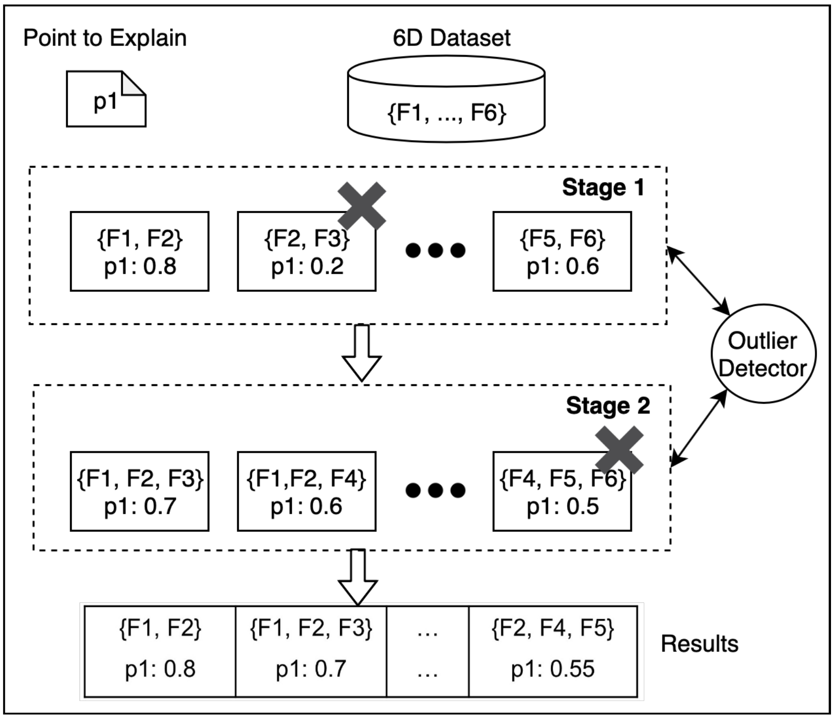
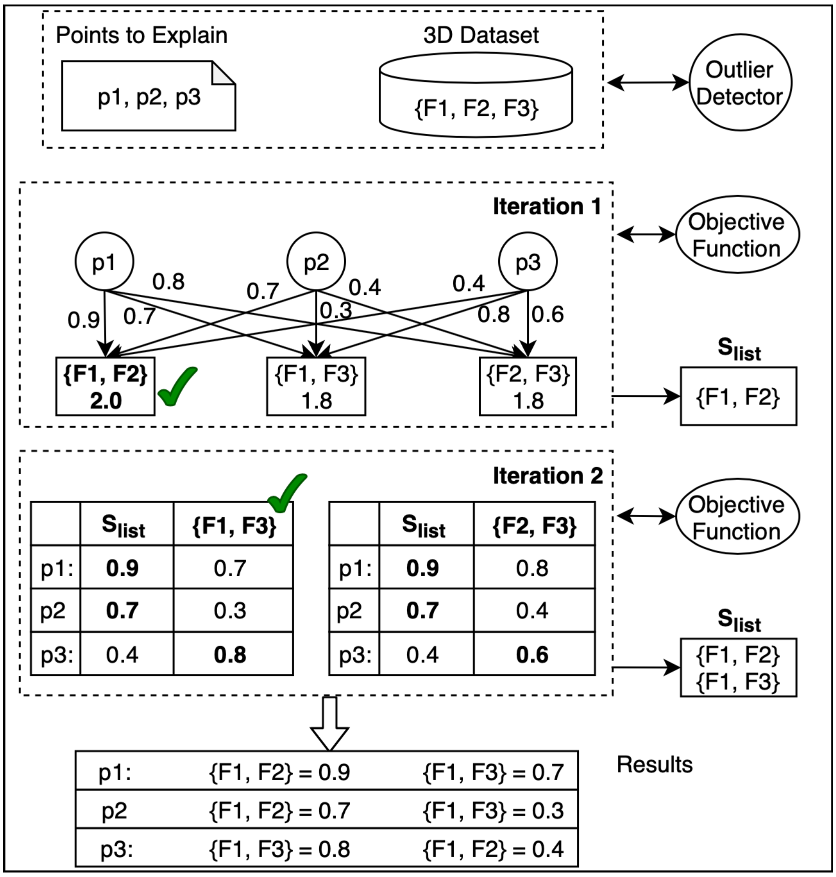


Figure 4.2: Beam steps to find subspaces up to 3 dimensions from a $6d$ dataset to explain the point p1

explore feature subspaces in a dataset. The main difference is that the *outlyingness criterion* is applied *collectively* for all outliers rather than individually. The additional challenge stems from the fact that some outliers may be explained by subspaces of different dimensionality or in an extreme case all outliers could be explained by different subspaces.

LookOut [29] searches *exhaustively* subspaces of *fixed* dimensionality and returns those that exhibit a certain *utility*. LookOut was genuinely used to obtain $2d$ subspaces that can be easily visualized in order to explain a set of outliers. However, we used the algorithm to explore subspaces of high dimensionality as well. LookOut formalizes explanation summarization as maximization problem using an objective function equipped with the following properties: (i) *non-negative* , (ii) *non-decreasing* and iii) *sub-modular*. As submodular optimization is known to be an **NP-hard** problem, greedy approximation techniques are used (e.g., with a 63% approximation guarantee [59]). The main algorithmic steps of LookOut are depicted via an example in Figure 4.3. Given (i) a set of outlier points $P = \{p1, p2, p3\}$ and (ii) a number of top-k explanation summaries (i.e., the budget of the computation), LookOut constructs a subspace list S_{list} with the top-k subspaces that maximize the scores of the three points i.e., they provide a concise summary. Initially, LookOut employs an off-the-self outlier detector to score all outliers in the three possible $2d$ subspaces drawn from the $3d$ feature space of



	S_{list}	{F1, F3}
p1:	**0.9**	0.7
p2	**0.7**	0.3
p3:	0.4	**0.8**

	S_{list}	{F2, F3}
p1:	**0.9**	0.8
p2	**0.7**	0.4
p3:	0.4	**0.6**

p1:	{F1, F2} = 0.9	{F1, F3} = 0.7
p2	{F1, F2} = 0.7	{F1, F3} = 0.3
p3:	{F1, F3} = 0.8	{F1, F2} = 0.4

Figure 4.3: LookOut steps to find $2d$ subspaces from a $3d$ dataset with budget b = 2 (bold values indicate the highest scores per table row)

the dataset. LookOut's objective function for concise summarization is defined as follows:

$$f(S_{list}) = \sum_{p_i \in P} \max_{s_j \in S_{list}} score_{i,j}$$

where $score_{i,j}$ represents the outlier score that point p_i received in subspace s_j. Then, to assess utility of a subspace s to the S_{list}, LookOut examines its marginal gain computed as:

$$\Delta_f(s|S_{list}) = f(S_{list} \cup s) - f(S_{list})$$

In our example of Figure 4.3, S_{list} is initially empty and subspace $\{F1, F2\}$ is inserted during the first iteration as all three points obtain their best outlyingness score in this subspace. During the second iteration, LookOut examines which of the two remaining subspaces $\{F1, F3\}$ and $\{F2, F3\}$ provide the greatest marginal gain for S_{list}. In our example, $\{F1, F3\}$ has a higher marginal gain than $\{F2, F3\}$

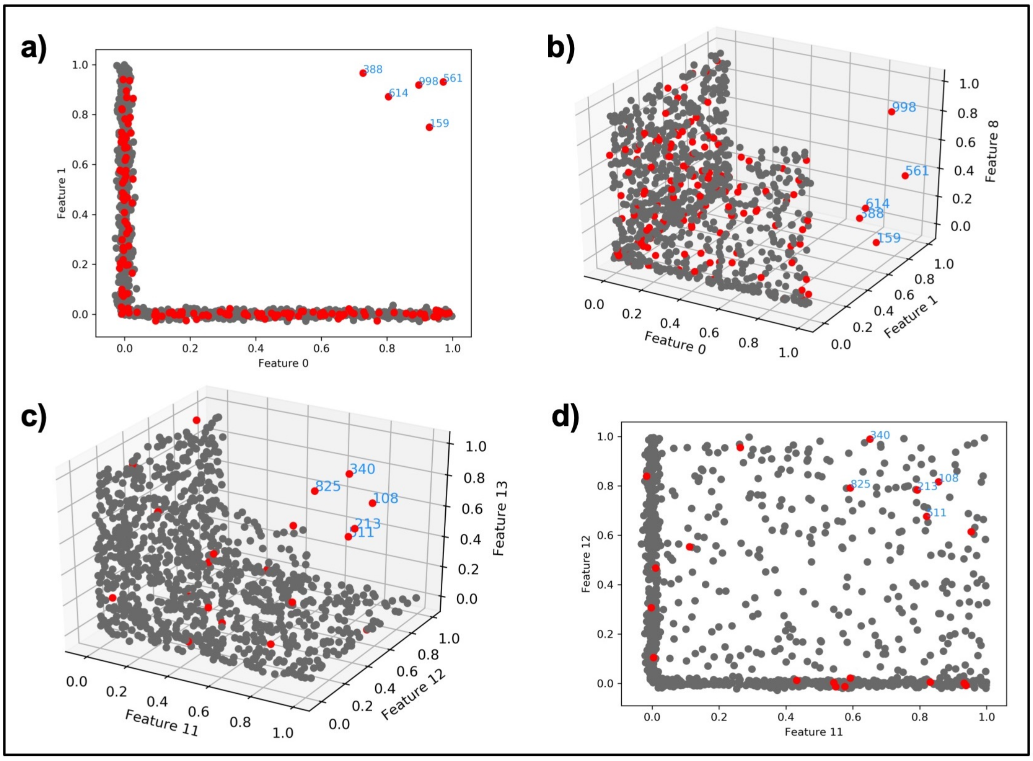


Figure 4.4: Data distribution in augmented/projected subspaces of HiCS Datasets

as its maximizes $p3$'s score, while $p1$ and $p2$ scores are already maximized by $\{F1, F2\}$. The two subspaces are compared w.r.t. the maximum scores of every point currently in S_{list}. As the budget in our example is 2 i.e., the number of subspaces that will be included in explanation, the process stops and the S_{list} is returned as a summary of the subspaces explaining the points given as input.

In a nutshell, LookOut returns the top-k subspaces of fixed dimensionality that concisely explain multiple outliers. A subspace is considered a good summary candidate at a certain iteration step if it maximizes the overall score for at least one outlier. Hence, LookOut's effectiveness strongly depends on the ability of an off-the-self outlier detector to highly score outliers in their relevant subspaces.

High Contrast Subspaces (HiCS) [36] relies on a subspace search strategy that exploits combinations of correlated features called high contrast subspaces. The underlying intuition is that high contrast subspaces have many empty regions and few very dense regions, thus they are good candidates for separating outliers from inliers. Figures 4.4-a) to -c) illustrate three subspaces with correlated features ($\{F0, F1\}$, $\{F0, F1, F8\}$ and $\{F11, F12, F13\}$) while Figure 4.4-d) a subspace with non correlated features ($\{F11, F12\}$). Subspace contrast in HiCS is measured using two-sample statistical tests[2] which are applied to the raw feature values

[2] The Welch's t-test or the Kolmogorov-Smirnov test.

under the null hypothesis that *both samples originate from the same underlying probability density function.* To enhance statistical precision, HiCS performs the statistical test for several Monte Carlo iterations and the average score is computed per subspace.

HiCS searches for high contrast subspaces via a stage-wise technique. In the first stage, it scores exhaustively all the $2d$ subspaces and selects the top-k based on their contrast. In next stage, the best $2d$ subspaces, are used to construct $3d$ subspaces scored again based on their contrast. The same procedure is repeated for several stages until reaching the full feature space d of a d-dimensional dataset; hence, the algorithm may retrieve subspaces of varying dimensionality. HiCS has been originally evaluated with LOF, but in principle any other off-the-self detector could be employed. In order to make a fair comparison with LookOut, we force HiCS to return subspaces of fixed dimensionality up to a predefined stage. We call this variation $HiCS_{FX}$.

To conclude, HiCS is a best effort algorithm that exploits subspaces with correlated features to discover summaries of varying dimensionality. Although the assumption that outliers are more likely to appear in combinations of correlated features seems effective for highly clustered anomalies, correlated subspaces may not always explain outliers, as depicted in Figure 1.1-e). The main novelty of HiCS lies in the decoupling of the subspace search strategy from the scores assigned by an off-the-self detector to a set of outliers.

Chapter 5

Benchmarking Environment

The algorithms along with the datasets used in our testbed are available in our GitHub repository[1] to ensure repeatability of our experiments. Regarding outlier detectors, we used the implementation of LOF and iForest from Scikit-learn [64] and Fast ABOD from PyOD [95]. We have implemented LookOut, RefOut and Beam in java and modified HiCS implementation from ELKI [71]. Our primary concern in this work is the correctness of the implemented explanation algorithms. All experiments were performed in a Windows personal computer with a 4 core Intel i7 processor and 16GB of main memory.

5.1 Pipelines of Executed Algorithms

As illustrated in Figure 5.1, given (i) a dataset, (ii) a set of outliers (points of interest) and (iii) a target dimensionality to explain them, we execute all the possible pairs of explanation and detection algorithms. Each executed pipeline results to a list of *fixed*-dimensionality subspaces considered as relevant to each point of interest. The effectiveness of each pipeline is assessed using the relevant subspace(s) per point available in the ground truth of each dataset and the metric that we define in Section 5.3.

Regarding the hyper-parameters of the outlier detectors, we ensure that they are able to identify all the outlier points in their relevant subspace(s) provided in the ground truth. For LOF we use $k = 15$ and for Fast_ABOD $k = 10$. We run iForest for 10 repetitions to reduce the variance of outlyingness scores and the average score is computed for every point, using $t = 100$ trees and $sub - sample\ size = 256$. Regarding the hyper-parameters of the explainers, for HiCS we use $candidateCutOff = 400$, $a = 0.1$, $Monte\ Carlo\ Iterations = 100$ and Welch's t-test is performed. For LookOut we use $budget = 100$. For Beam we use $beam - width = 100$. For RefOut we use $poolsize = 100$, $beam - width = 100$, the random subspace dimensionality is set to 70% of dataset's dimensionality and

[1] `https://git.io/Jvu06`

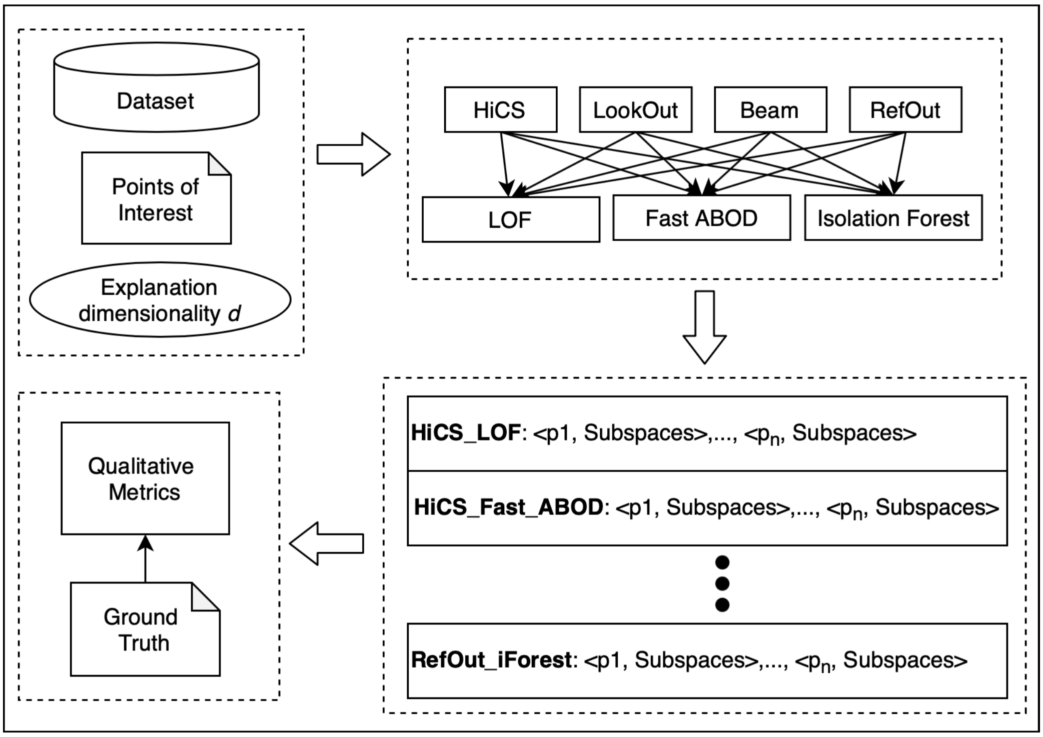

Dataset
Points of Interest
Explanation dimensionality *d*
HiCS
LookOut
Beam
RefOut
LOF
Fast ABOD
Isolation Forest
HiCS_LOF: <p1, Subspaces>,..., <p$_n$, Subspaces>
HiCS_Fast_ABOD: <p1, Subspaces>,..., <p$_n$, Subspaces>
RefOut_iForest: <p1, Subspaces>,..., <p$_n$, Subspaces>
Qualitative Metrics
Ground Truth

Figure 5.1: Pipelines of outlier detectors & explainers

Welch's t-test is performed. For HiCS, Beam and RefOut we return the top-100 subspaces as the final result.

5.2 Real and Synthetic Datasets

In this section we describe the real and synthetic datasets used in our testbed. To reduce confounding factors in the experimental evaluation of the algorithms, the selected datasets are mainly contaminated with *density-based* outliers. Outliers of this type can be detected by LOF but under certain conditions also by other detectors like ABOD and iForest (see Section 6). The main characteristics of our

Characteristics	Real Datasets (# 3)	Synthetic Datasets (# 5)
Outlier Type	Full Space	Subspace
Explanation Dimensionality	2-4 *d*	2-5 *d*
% Contamination with Outliers	10%	2, 3.4, 5.9, 10, 14.3 %
# Relevant Subspaces	60 (A), 151 (B), 249 (C)	4, 7, 12, 22, 31
# Relevant Subspaces per Outlier	3 (1 per dimensionality)	1 (91% outliers), 2 (9% outliers)
# Outliers per Relevant Subspace	1 (A), 1.13 (B), 1.45 (C)	5
% Relevant Feature Ratio	100%	35, 21, 12, 7, 5 %
Outlier Visibility w.r.t. Relevant Subspaces	Projections / Augmentations	Augmentations

Table 5.1: Characteristics of real and synthetic datasets

datasets are summarized in Table 5.1. To compute the relevant feature ratio for *subspace outliers* we took the fraction of the highest dimensional relevant subspace that explains a portion of the outliers over the total number of features in the dataset. Note that for *full space* outliers this ratio is 100% as all features are considered relevant.

Breast, Breast Diagnostic and *Electricity Meter* are real-world datasets widely used to benchmark ML methods for anomaly detection [19]. To facilitate comparison with already published results, we used the version of these datasets[2] made available by the authors of RefOut algorithm [38]. Specifically, Breast (A) contains 198 points, 31 features and 20 outliers, Breast Diagnostic (B) contains 569 points, 30 features and 57 outliers and Electricity (C) contains 1205 samples, 23 features and 121 outliers. The ground truth provided per dataset contains the outliers detected by LOF resulting 10% contamination with outliers. Note that the experiments in [38] revealed that the reported outliers are *full space*. To obtain the best subspaces explaining them[3], we followed the method as described in [38] by performing an exhaustive search from 2 up to 4 dimensions for every dataset using LOF and keeping the top scored subspace per outlier at the corresponding dimension. We started from 2 dimensions as the initial step of HiCS and Beam perform an exhaustive search in $2d$ subspaces. We should stress that outliers are identifiable by LOF in both *lower dimensional projections* and *augmentations* (i.e., supersets) of the relevant subspaces. These datasets are challenging for summarization algorithms (HiCS and LookOut) as the relevant subspaces can best explain one outlier on average, e.g., for Electricity there are 1.43 outliers explained per relevant subspace (see Table 5.1).

HiCS synthetic datasets[4] were created by the authors of the HiCS [36] algorithm featuring *subspace outliers*. They initially splitted the datasets into $2d$ up to $5d$ subspaces, and generated high density clusters in each subspace. Then, they randomly picked 5 points and modified them to deviate from all clusters in each subspace. From these datasets we picked the dataset with the maximum dimensionality ($100d$) and splitted it into five sub-datasets from 14 up to 100 dimensions. The ratio of relevant features is depicted in Table 5.1 ordered from low ($14d$) to high ($100d$) number of features. Note that every dataset contains 1000 points. As illustrated in Figure 5.2 and Table 5.1, this split produced datasets of increasing (i) data dimensionality (i.e., number of features), (ii) number of relevant subspaces of different dimensionality and (iii) contamination with outliers. In HiCS datasets, the relevant subspaces and the outliers were given but there was no association between them. To identify the relevant subspace per outlier, we run LOF and keep the top-5 outliers with the highest scores per relevant subspace. The so obtained ground truth is aligned with the original contamination of the dataset with 5 points deviating in each relevant subspace that can be easily detected by LOF.

[2] `https://www.ipd.kit.edu/~muellere/RefOut/`

[3] We discovered that the subspaces originally reported by the authors of RefOut were not optimal for most outliers.

[4] `https://www.ipd.kit.edu/~muellere/HiCS/`

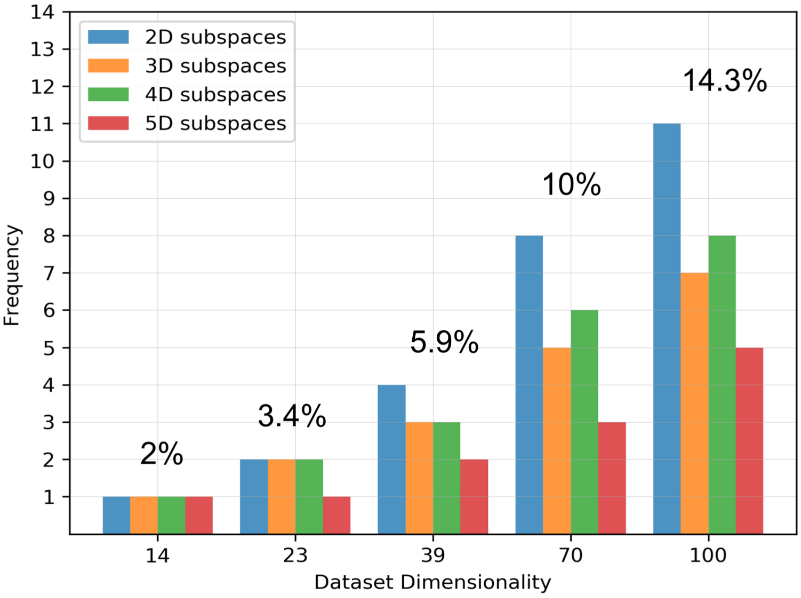


Figure 5.2: Dimensionality of subspaces relevant to outliers and contamination ratio of HiCS datasets

An example of a $2d$ and a $3d$ relevant subspace is illustrated in Figures 4.4-a) and -c).

Note that the vast majority ($\sim 91\%$) of outliers in HiCS datasets is explained by one subspace and few outliers ($\sim 9\%$) by two different subspaces. These subspaces follow the properties: (i) they are disjoint in terms of features, (ii) each subspace can explain exactly five outlier points, (iii) they have highly correlated features, (iv) outliers are identifiable by the detectors in *augmented subspaces*, i.e., supersets of the relevant features (see example of Figures 4.4-a) and -b) and (v) outliers are mixed with inliers in *lower dimensional projections* of relevant subspaces (see example of Figures 4.4-c) and -d). Note that all outliers in HiCS datasets can be discovered by the three detectors used in our testbed.

5.3 Evaluation Metric

In this section we present the metric used to evaluate the effectiveness of the 12 pairs of outlier detection and explanation algorithms (see Figure 5.1). Although outlier explanations target human analysts, we have not conducted user studies as our datasets are equipped with ground truth regarding which subspaces are relevant to the outliers they contain.

We denote the set of points of interest as P, the set of the relevant subspaces per point $p \in P$ as REL_p, and the returned subspaces from an explanation algorithm a to a point p as $EXP_a(p)$. As each outlier in our datasets has very few relevant subspaces (specifically 1-3), we selected the MAP metric penalizing detectors that do not rank the relevant subspace(s) for an outlier within the top positions [77]. To compute MAP of an explainer a for a set of points P, we initially compute the precision (see Eq. 5.1) which is used to compute the Average Precision (see Eq. 5.2). $P@k(p)$ denotes the precision up to a k-th position of the returned subspaces in $EXP_a(p)$. The Boolean function $rel(k)$ indicates whether a subspace at the k-th position of $EXP_a(p)$ is relevant or not. Then, MAP is computed using the Average Precision of all points explained at a given dimensionality (see Eq. 5.3) according to the ground truth. A high MAP value indicates that for several points, the explainer was able to find and highly score their relevant subspaces using an outlier detector. Compared to other metrics such as accuracy, precision or recall, MAP better captures the scoring nature of outlier explanation algorithms: the discovered relevant subspaces should be ranked at the top positions of the list of candidates an algorithm considers.

$$\text{Precision}_a(p) = \frac{|REL_p \cap EXP_a(p)|}{|EXP_a(p)|} \tag{5.1}$$

$$\text{AveP}_a(p) = \frac{\sum_{k=1}^{|EXP_a(p)|} \text{P@k}(p) * rel(k)}{|REL_p|} \tag{5.2}$$

$$\text{MAP}_a(P) = \frac{1}{|P|} \sum_{p \in P} \text{AveP}(p) \tag{5.3}$$

Chapter 6

Experiments and Insights

In this section we present our experiments for comparing point explanation and summarization algorithms. Our testbed includes the real datasets used in the evaluation of RefOut [38] as well as the synthetic datasets used in the evaluation of HiCS [36]. Both types of datasets were originally used to assess the effectiveness of detecting outliers hidden in subspaces rather than the suitability of the subspaces that led to the detection of those outliers. To the best of our knowledge, the only study investigating recall and precision of the subspaces of varying dimensionality retrieved by Beam was presented in [62]. In our testbed, we measure the *effectiveness* and *efficiency* of the four explanation algorithms when seeking for explanations of increasing dimensionality ($2d$ up to $5d$) as this experiment reveals useful insights when different outlier detectors are used.

6.1 Evaluation of Point Explanation Algorithms

The experiments of this section aim to answer two questions: (a) Is it effective to combine any explanation algorithm with any off-the-shelf outlier detector? (b) How is the behavior of outlier detection and explanation pipelines affected by the number of features in a dataset? To answer these questions, we run Beam and RefOut with LOF, Fast ABOD and iForest using the settings described in Section 5.1 for the synthetic and real-world datasets presented in Section 5.2. Figure 6.1 depicts for each dataset, the MAP (y-axis) of different outlier detection and explanation pipelines for explanations of increasing dimensionality (x-axis).

Figures 6.1-a) to -e) illustrate the MAP obtained in the five synthetic datasets of our testbed. Starting from the 14 dimensions in Figure 6.1-a), we observe that RefOut with LOF achieves optimal MAP as it retrieves and gives the highest score to the relevant subspaces for all the outliers, regardless of the explanation dimensionality. This is because (i) HiCS datasets contain highly clustered anomalies, thus LOF is the most suitable detector and (ii) the pool of RefOut contains low dimensional subspaces in which outliers can be more easily detected. Note that Beam with LOF has lower MAP for high explanation dimensionality since it does

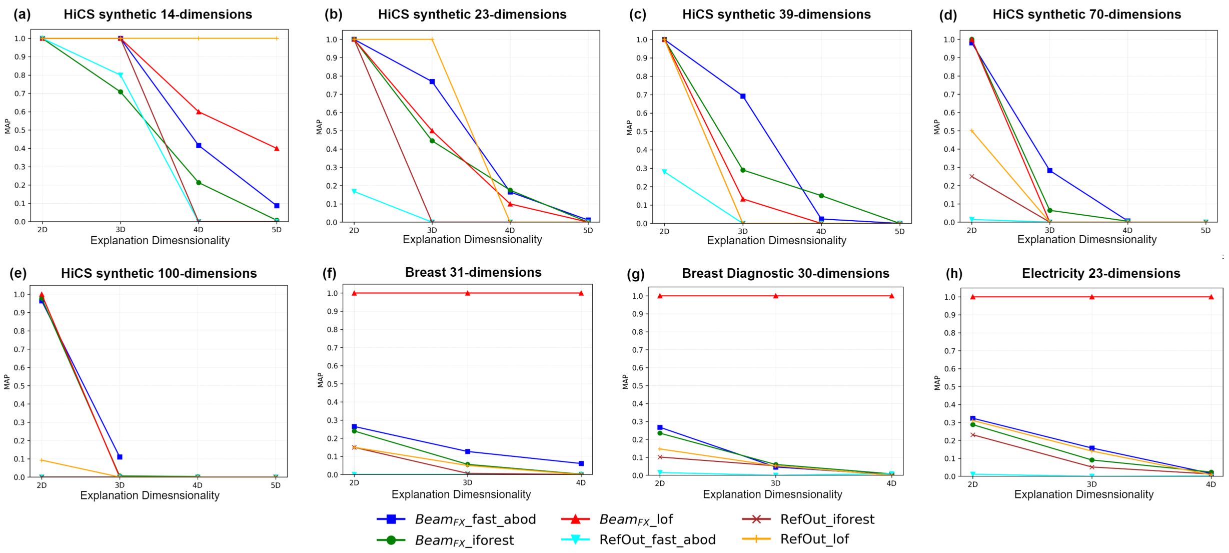

(a) HiCS synthetic 14-dimensions
(b) HiCS synthetic 23-dimensions
(c) HiCS synthetic 39-dimensions
(d) HiCS synthetic 70-dimensions
(e) HiCS synthetic 100-dimensions
(f) Breast 31-dimensions
(g) Breast Diagnostic 30-dimensions
(h) Electricity 23-dimensions

Figure 6.1: Mean Average Precision (MAP) of Beam and RefOut in HiCS synthetic datasets (a)-(e) and real-world datasets (f)-(h) for explanations of increasing dimensionality (best viewed in color)

not retrieve all the relevant subspaces. Passing to 23 dimensions in Figure 6.1-b), the effectiveness of every pipeline drops especially for high dimensional explanations. RefOut with LOF seems to not be affected up to $3d$ explanations. An interesting behavior observed in this plot is that Beam is more effective with Fast ABOD and iForest than with LOF. This is due to the fact that the stage-wise strategy of Beam requires to collect lower dimensional projections of the relevant subspaces, so they could be formed in the final stage. Recall that in HiCS datasets, outliers are not separated from inliers in lower projections of the relevant subspaces (see Figure 4.4). According to complementary experiments not presented here due to space restrictions, in the early stages of Beam, the score distributions of outliers and inliers overlap less when Fast ABOD and iForest is used instead of LOF.

While the dimensionality of datasets increases, the same trends are observed in Figures 6.1-c) to -e). In general, Beam is able to retrieve all relevant $2d$ subspaces with the three detectors due to the exhaustive scoring of all feature pairs. However, its effectiveness starts dropping when the dimensionality of explanations increases. As the number of Beam stages increase, more subspaces need to be collected stage-wise with smaller differences in their score. RefOut proves to be more sensitive than Beam w.r.t. the number of features in the dataset D. As the dimensionality of random subspace projections in the pool is proportional to D's dimensionality, it becomes more difficult for RefOut to identify important features due to the less distinguishable score populations in subspaces. Observe that none of the algorithms seem to work for $4d$ explanations from 70 dimensions and higher and for $5d$ explanations from 23 dimensions and higher. Note that we run 10 times iForest

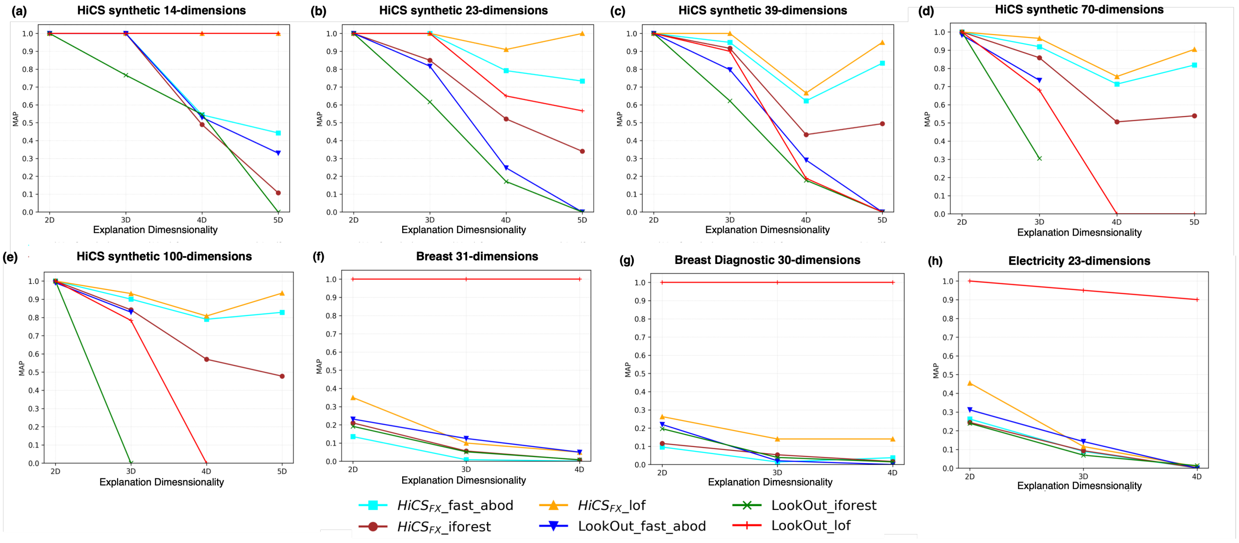


Figure 6.2: Mean Average Precision (MAP) of HiCS and LookOut in HiCS synthetic datasets (a)-(e) and real-world datasets (f)-(h) for explanations of increasing dimensionality (best viewed in color)

(see Section 5.1) for every subspace considered by Beam up to $4d$ explanations for $70d$ and $100d$ datasets and Fast ABOD up to $4d$ explanations in $70d$ and up to $3d$ in $100d$ datasets. Specifically, to explain 100 outliers with $5d$ explanations in a $70d$ dataset, Beam needs to assess approximately 2.2M subspaces. In Section 6.3, we demonstrate that Beam requires an efficient detector such as LOF to assess a significant amount of subspaces.

Figures 6.1-f) to -h) illustrate the MAP obtained in the three real-world datasets of our testbed. Recall that in these datasets, the majority of the outliers are identifiable even in the full feature space. In general, Beam with LOF retrieves the optimal subspace for every outlier point (MAP = 1), despite of the explanation dimensionality. However, the effectiveness of Beam with Fast ABOD and iForest is significantly lower. On the contrary, RefOut seems to have very low MAP regardless of the employed detector. This is because RefOut cannot distinguish which features of full space outliers affect significantly the score populations generated by the corresponding detector.

Lessons Learned. Depending on the dataset characteristics, outlier detectors behave differently, affecting the effectiveness of explanation algorithms. A critical factor is whether outliers are masked by inliers in lower dimensional projections of the relevant subspaces (as in HiCS datasets). In this case, for datasets and explanations of low dimensionality, RefOut's random projection technique along with a detector suitable for the nature of outliers (e.g., LOF for clustered outliers) is preferred. For high dimensional datasets and low explanation dimensionality, Beam's stage-wise technique along with iForest or ABOD can effectively capture the small deviation of outliers in the subspaces considered by early stages. None of the algorithms seems to work for high explanation dimensionality (e.g., $4d$ and $5d$) and high dataset dimensionality (e.g., $70d$ and $100d$). When outliers are also visible in the full feature space (as in real-world datasets) the random projection technique exhibits poor MAP as it fails to find relevant features that significantly affect the score distributions. In this case, a stage-wise technique coupled with a suitable detector should be preferred regardless of the explanation dimensionality.

6.2 Evaluation of Summarization Algorithms

The experiments presented in this section aim to answer three questions: (a) Is it effective to combine any explanation summarization algorithm with any outlier detector?, (b) How is the behavior of outlier detection and explanation pipelines affected by the number of features or their correlation in a dataset?, and (c) What is the quality of summaries in the presence of outliers explained by subspaces of different dimensionality? To answer these questions, we run HiCS and LookOut with LOF, Fast ABOD and iForest using the settings described in Section 5.1 for the synthetic and real-world datasets presented in Section 5.2. Figure 6.2 depicts per dataset the MAP (y-axis) of different pairs of outlier detection and explanation algorithms for explanations of increasing dimensionality (x-axis). Despite the fact that HiCS does not use any detector to search candidate subspaces, it employs a detector to rank the retrieved subspaces. Thus, its effectiveness should be also evaluated for different detectors.

Figures 6.2-a) to -e) show the MAP of different algorithms for the five synthetic datasets of our testbed. Starting from 14 dimensions in Figure 6.2-a), HiCS and LookOut with LOF achieve optimal MAP regardless of the explanation dimensionality. As dataset's dimensionality and outlier ratio increase in Figures 6.2-b) to -e), HiCS with LOF and Fast ABOD are the most effective because (i) small groups of outliers are hidden within subspaces with correlated features and (ii) outliers are highly clustered at the borders of data distribution, allowing LOF and Fast ABOD to score their relevant subspaces at the top positions. The lowest MAP value of HiCS is observed in the 39 dimensional dataset where some $4d$ relevant subspaces do not contain highly correlated features. This drop clearly demonstrates the strong dependency of HiCS on the feature correlation heuristic.

As we can see in Figures 6.2-b and -e) LookOut's effectiveness significantly drops as the explanation dimensionality increases in higher dimensional datasets. One reason of this drop is related to the lower scores returned by the detectors in high dimensional subspaces. An additional reason stems from the existence of points exhibiting high outlyingness also in their augmented subspaces. According to complementary experiments, detectors (especially LOF and iForest) assign higher scores to outliers in their augmented subspaces of dimensionality d than to outliers explained exclusively in d. As the outlier ratio increases along with dataset's dimensionality, more outliers get high scores in their augmented subspaces of a requested dimensionality. As a small fraction of outliers is explained by high dimensional subspaces, LookOut mainly retrieves augmented subspaces of outliers explained in lower dimensions that provide higher marginal gain. Observe that LookOut with Fast ABOD starts performing better than with LOF for high dataset dimensionality. Note that we run LookOut with LOF up to $4d$ explanations in 100 dimensions and Fast ABOD and iForest only up to $3d$ explanations for 70 and 100 dimensions. Specifically, to explain the outliers with $4d$ explanations in a $70d$ dataset, LookOut needs to assess 900K subspaces. In Section 6.3, we demonstrate that LOF is the most efficient detector when a significant amount of subspaces need to be assessed.

Figures 6.2-f) to -h) illustrate the MAP obtained in the three real-world datasets of our testbed. HiCS has poor MAP regardless of the explanation dimensionality or the detector used. This is because outliers are not contained in subspaces with highly correlated features. LookOut with LOF (used to identify the outliers) is the most effective as it is able to retrieve almost all relevant subspaces even when they maximally explain one outlier. On the contrary, LookOut with iForest and Fast ABOD exhibit poor performance as they are not able to highly score the relevant subspaces.

Lessons Learned. The fact that relevant subspaces may be formed by highly correlated features could be exploited to avoid a blind search of subspaces. When datasets exhibit strong feature correlation in relevant subspaces, HiCS exploits this heuristic and provides the best performance regardless of the dataset's or explanation's dimensionality. It only depends on the ability of LOF or Fast ABOD to highly rank the retrieved subspaces. LookOut is as effective as HiCS in the synthetic datasets for low dataset dimensionality (e.g. $14d$). When subspaces are formed by uncorrelated features, LookOut is a better alternative. However, LookOut is heavily impacted by the varying dimensionality of subspaces explaining different outliers. Indeed, the utility of subspaces in LookOut is defined exclusively in terms of their scores, without considering any semantic property of explanations such as the coverage of the points to be explained, or the overlap or the equivalence of subspaces in the explanation summaries.

6.3 Algorithms RunTime & Tradeoffs

In this section we report the execution time of the two point explanation and the two summarization algorithms we evaluated their effectiveness in Sections 6.1 and 6.2. In this respect, we are using the same synthetic (up to HiCS $39d$) and real (Electricity $23d$) datasets containing a similar amount of samples (~ 1000). We demonstrate the execution time only for Electricity real dataset as it contains the highest number of samples exhibiting the same behavioral trends as the other two real datasets. Recall that as we are looking for explanations of fixed dimensionality ($2d$ up to $5d$) the ratio of relevant features decreases as dataset's dimensionality increases.

Outlier Detection. Unlike HiCS, subspace search in explanation algorithms like Beam, RefOut and LookOut, heavily depends on the efficiency (and effectiveness) of used off-the-self detectors. According to the performance curves of detection and explanation pipelines depicted in Figure 6.3, LOF is the fastest followed by iForest and Fast ABOD across all datasets and explanation algorithms. This is due to low number of samples (~ 1000) despite the fact that iForest has the lowest time complexity. A similar result has been reported in [18] for the same hyper-parameter settings as those used in our testbed (see Section 5.1). Note that for iForest we report the average time out of 10 repetitions per subspace. Specifically, to score a single subspace LOF needed 0.05, iForest 0.2 and Fast ABOD 2 seconds approximately.

Point Explanation. The runtime of pipelines involving Beam, RefOut are illustrated in Figures 6.3-a) to -d). Critical factors affecting Beam's efficiency are: (i) the requested explanation dimensionality (more stages to be built), (ii) the dataset's dimensionality (more subspaces to be assessed per stage), (iii) the efficiency of the employed detector and (iv) the number of outliers to explain (the process is repeated per outlier). However, due to its random sampling technique, RefOut's runtime is relatively stable regardless of the explanation or dataset's dimensionality. Note that up to $39d$ datasets and $2d$ explanations, RefOut and Beam with LOF need almost the same time to assess a similar amount of subspaces. RefOut with LOF outperforms Beam with LOF from 1 (in real datasets) up to 3 orders (in synthetic datasets) of magnitude for $39d$ datasets and $5d$ explanations.

Explanation Summarization. The runtime of pipelines involving LookOut and HiCS are illustrated in Figures 6.3-e) to -h). The critical factors affecting LookOut's efficiency are: (i) dataset's and explanation dimensionality (exhaustive subspace search) and (ii) the efficiency of the employed detector. On the other hand, by decoupling subspace search from outlier scoring, the critical factor of HiCS efficiency is only the explanation dimensionality (more subspaces to be assessed per stage). Thus, HiCS exhibits similar running times when executed with LOF, iForest and Fast ABOD (used only to rank the discovered subspaces). Surprisingly, LookOut with LOF[1] outperforms all HiCS pipelines up to $4d$ explanations (by 1 order of magnitude in $2d$). For the size of datasets used in our experiments, HiCS statistical tests to assess feature correlation prove to be more costly than LOF distance calculation of points to assess their outlyingness. Performance gains of LookOut with LOF drop as we increase the number of features along with explanation dimensionality, leading HiCS to outperform LookOut in the $39d$ dataset for $5d$ explanations.

Table 6.1 demonstrates the point explanation and summarization algorithms along with their corresponding detector that exhibit the best tradeoff between effectiveness (according to Figures 6.1 and 6.2) and efficiency (according to Figure

[1] LookOut has been experimentally evaluated by its authors [29] only with iForest and $2d$ explanations.

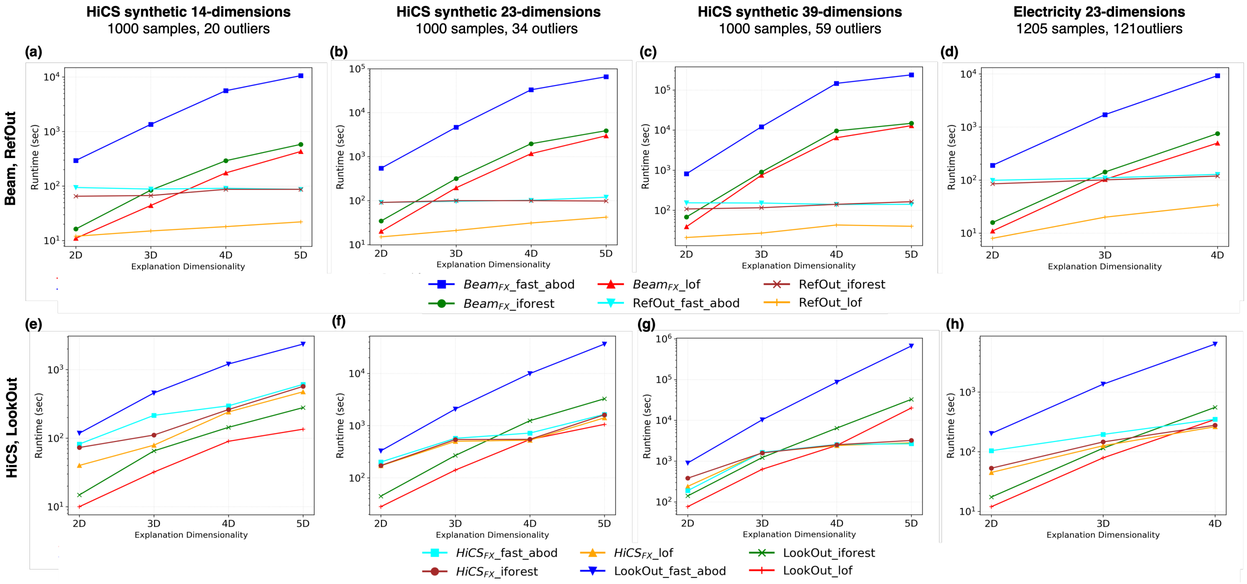


Figure 6.3: Runtime of detection and explanation pipelines (best viewed in color)

6.3) from $2d$ up to $5d$ explanations across decreasing relevant feature ratios. For every cell we take the top pair of algorithms according to their efficiency and effectiveness in pareto order. We prioritize generic algorithms like LookOut over algorithms like HiCS that work under specific conditions. For instance, LookOut with LOF is slightly less effective than HiCS with LOF in Figure 6.2-c), while they have the same execution time in Figure 6.3-g). In cells $2d$ and $3d$ with a 12% ratio, we consider that LookOut achieves a better tradeoff since it is more generic than HiCS. When point explanation or summarization algorithms exhibit zero effectiveness in all executed pipelines for a particular dataset and explanation dimensionality, no top pair is reported. For instance, for $5d$ and 21% or 12% ratios only one pair for detection and summarization algorithms is reported (HiCS with LOF) as no point explanation algorithm succeeds to return relevant $5d$ explanations. The main conclusions drawn from Table 6.1 are:

1. *State-wise subspace search* employed by Beam achieves the best tradeoff for full space outliers. Both its effectiveness and efficiency significantly decrease for subspace outliers as the ratio of relevant features decreases. However, it is the only option for high explanation dimensionality ($3d$ - $4d$) and low relevant feature ratio (< 12%).

2. *Random subspace projection* employed by RefOut provides a good tradeoff for subspace outliers with a medium ratio of relevant features (35% and 21%). Its effectiveness drops to zero as the explanation dimensionality becomes greater than $3d$ (for 21% ratio).

3. *Exhaustive subspace search* employed by LookOut exhibits top effectiveness and efficiency for full space outliers regardless of the explanation dimensionality,

Explanation Dimensionality	**Relevant Features Ratio**			
	100%	**35%**	**21%**	**12%**
$2d$	Beam LOF LookOut LOF	RefOut LOF LookOut LOF	RefOut LOF LookOut LOF	RefOut LOF LookOut LOF
$3d$	Beam LOF LookOut LOF	RefOut LOF LookOut LOF	RefOut LOF LookOut LOF	Beam Fast Abod LookOut LOF
$4d$	Beam LOF LookOut LOF	RefOut LOF LookOut LOF	Beam iForest HiCS LOF	Beam iForest HiCS LOF
$5d$	Beam LOF LookOut LOF	RefOut LOF LookOut LOF	HiCS LOF	HiCS LOF

Table 6.1: Tradeoffs of outlier detection and explanation algorithms

as well as, for subspace outliers up to $3d$. Its effectiveness significantly drops for subspace outliers explained by subspaces greater than $3d$ (for 21% ratio).

4. Correlation heuristic exploited by HiCS achieves the best tradeoff for $4d$-$5d$ explanations especially when the relevant feature ratio is low. This heuristic however, strongly depends on the data distribution as highly clustered outliers may are not always be visible in correlated features.

Part II

PROTEUS: Predictive Explanation of Anomalies

Chapter 7

PROTEUS AutoML pipelines

Figure 7.1 illustrates the main steps of the pipelines automatically generated by PROTEUS. We proceed with explaining each step as well as the underlying design choices.

Producing Predictive Explanations as a Supervised Task. First, the anomaly detector runs in dataset D for producing the anomaly scores which are then transformed into binary labels (anomaly or not) in dataset D'. Producing a surrogate model of lower dimensionality becomes a supervised, binary classification task with feature selection, where the outcome is the label of the unsupervised detector. We note that *data are standardized* for subsequent steps so that the standard deviation of each feature is 1.

Oversampling. D' is expected to be highly imbalanced (w.r.t. the outcome), as anomalies are rare. Imbalanced datasets are statistically challenging for any ML classifier. One technique to alleviate the problem is *oversampling* the minority class. We focus on *synthetic minority oversampling*, i.e., the samples are perturbed by adding noise to the values of the features, creating new points called *pseudo-samples.* In standard (unsupervised) oversampling, for small enough perturbations the pseudo-samples are *assumed* to remain in the minority class. An assumption that strongly depends on the definition of what is considered "small-enough". However, in this context, one can take advantage that the detector model produced in the first step is available to query regarding the label of a pseudo-sample. In other words, PROTEUS oversampling is *supervised.* Intuitively, oversampling probes the region around the anomalies and perturbs these points to examine if they cross the detector's decision boundary or not. It thus effectively increases the available sample size for the classification, potentially increasing the quality of the approximation with the surrogate model. For each anomalous sample a it produces ps pseudo-samples per anomaly by adding a perturbation vector p to a: $a' \leftarrow a+p$. Each p follows a multi-variate (d-dimensional) normal distribution with zero mean and an isotropic, diagonal, covariance matrix σI; σ is a hyper-parameter of the algorithm which we set to 0.1 for all the computational experiments. If a' is labelled as an anomaly (i.e., $\hat{Y}_A(a') = 1$) it is appended to the oversampled dataset

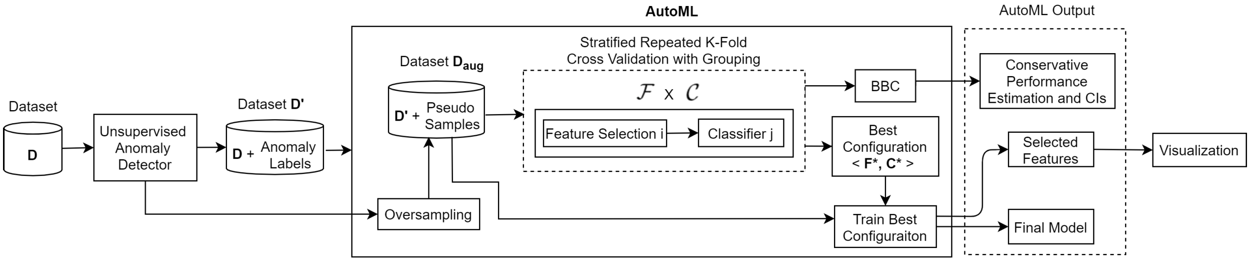


Figure 7.1: Proteus AutoML Pipeline for Anomaly Detection and Explanation

D_{aug}, otherwise, another pseudo-sample is produced.

Choosing the Performance Metric. The quality of performance of a predictive explanation requires metrics that are insensitive to the class distribution as anomalies are rare. In PROTEUS we optimize the area under the Receiver Operating Characteristic curve (**AUC**) which is a widely-used evaluation metric in anomaly detection, independent of the class distribution. Given a minimal subset of features and a classifier c, AUC equals the probability that an anomalous sample will get a higher score by c than an inlier. Discovering such minimal subset is a challenging task as the search space is exponential and features in the input dataset may be both *irrelevant* or *redundant* w.r.t. to the predictive outcome. As we describe later in this chapter, in this work we rely on effective and efficient *feature selection* algorithms [47, 86, 80] to extract predictive explanations.

Hyper-Parameter Optimization Space. To produce small-sized explanations, PROTEUS relies on feature selection algorithms, while to produce the surrogate model, a classifier is required. Most classification algorithms also accept a set of hyper-parameter values that also need to be tuned. We will call a combination of feature selection and classification algorithms and their hyper-parameters values as a *configuration. Each configuration is a pipeline that accepts a dataset and produces a classification model and corresponding selected features.* PROTEUS searches the configuration space for the one that leads to an optimal model by performing a simple grid search [33]. The search space of configurations is formed by the Cartesian product $\mathcal{F} \times \mathcal{C}$ (see Figure 7.1) where $\mathcal{F}$ ($\mathcal{C}$, respectively) is the set of all feature selection (classification) algorithms with bounded hyper-parameter values.

As our choices for feature selection algorithms, we include the Statistical Equivalent Signatures (SES) [47], Forward-Backward with Early Dropping (FBED) [86], and Lasso [80]. All of them guarantee to return the optimal feature subset (Markov Blanket in Bayesian Networks) under certain broad (but different for each algorithm) conditions, removing not only *irrelevant*, but also *redundant* features. In general, SES and FBED tend to return smaller feature subsets than Lasso, with a small drop in predictive performance [86].

Regarding the hyper-parameters of feature selection algorithms, for SES we used max conditioning set $k \in \{2, 3\}$, and significance threshold $a \in \{0.01, 0.05, 0.1\}$, we run FBED for $K \in \{0, 1, 3\}$ iterations with significance threshold in $\{0.01, 0.05, 0.1\}$ and for Lasso we used $\lambda \in \{0.001, 0.01, 0.1, 0.2\}$.

As the decision boundary is rarely linear, we consider linear as well as non-linear classifiers. Our selection considers two facts (a) the extensive experimental results of [17], (b) the fact that deep neural network architectures are almost certain to overfit in very low sample sizes, both in terms of total sample size and the size of the rare class. The present selection of classifiers comprises of: (i) Support Vector Machines where we used *linear*, *polynomial* of degree 2 and 3, and the *rbf* with $\gamma \in \{1, 2, 5\}$ kernels, and cost penalty parameter $C \in \{1, 5, 10\}$; (ii) Random Forest with *entropy* split criterion, number of trees in $\{100, 300, 500\}$ and minimum leaf size in $\{1, 2, 3\}$ and (iii) K-Nearest-Neighbors with $K \in \{5, 10, 15\}$. The number of pseudo-samples to create per anomaly, called *ps* is also tuned as a hyper-parameter taking values in $\{0, 3, 10\}$. Of course, additional classifiers and feature selection algorithms can be easily integrated in PROTEUS. In total, PROTEUS tried 600 configurations. Finally, as anomaly explanation targets human analysts, *we limit the number of features selected up to 10.* To select these 10 features, we rank them based on their score given by the corresponding feature selection algorithm (e.g., Lasso coefficients) and pickup the top-10.

Estimating Performance for Tuning. What is considered as the optimal configuration, out of all tried, is *the one that leads to models with the highest expected out-of-sample (unseen samples) predictive performance.* It is important to estimate this quantity accurately, i.e., with small variance. *A smaller variance of estimation increases the probability that the truly optimal configuration will be selected, and thus improves the quality of the final model.* Estimation is challenging when there are only few anomalies in the dataset. Indicatively, the synthetic dataset used in our experiments (see Section 8.1) contains 10 anomalies out of 867 samples.

To estimate the expected out-of-sample performance, PROTEUS employs a *Stratified, R-Repeated K-fold Cross Validation with Grouping* protocol. We now explain each part of the protocol. We assume that the reader is familiar with the Standard K-fold Cross Validation (**CV**, hereafter). The *Stratified CV* is a variant where the partitioning to folds is performed under the constraint that the distribution of the classes in each fold is approximately the same as the one in the full dataset [87]. Stratification reduces the variance of estimation for imbalanced data and classes with very few samples (*ibid*). To further reduce the variance of estimation we repeat the CV process multiple times R and take the average (*R-Repeated CV*). Multiple repeats reduce the variance component due to the stochasticity of the specific partitioning. Prior work has shown its benefits *ibid.* Finally, we come to *Grouping.* By CV with Grouping we indicate a variant of CV that handles grouped samples (a.k.a. as clustered samples in statistics, not to be confused with clustering of samples). These are samples that are not independently sampled and maybe correlated given the data distribution. Such samples are repeated measurements on the same subject, as an example. In our context, *an anomaly and its pseudo-samples are grouped*: information from a pseudo-sample in the training set *leaks* to predicting the corresponding anomaly in the test fold. To avoid information leakage, CV with grouping partitions to folds with the constraint that all samples of a group remain in the same fold. In our experiments, we set the number of folds $K = 10$ and the repeats $R = 5$. Hence, each application of the current version of PROTEUS trains $(K \cdot R \cdot \#\ \text{Configurations} + 1) \cdot ps = 90,003$ models.

Producing the Final Surrogate Model and Feature Subset. The final model is trained using all available samples (the full D_{aug}) with the best configuration found, denoted with $\langle F^{*}, C^{*} \rangle$ in Figure 7.1. This configuration also produces the final subset selection (anomaly explanation). The reasoning is that most algorithms (and hence, configurations) are expected to produce better quality models and improved feature selection with more available sample. The models trained during the CV are only employed for selecting the optimal configuration and providing estimates.

Estimating the Out-of-Sample Performance. We now consider how the performance estimate of the final model is produced. Let us assume that 1000 configurations are tried and the best found has a CV estimate of 0.90 AUC. Unfortunately, *the CV estimate of the best configuration is optimistic and should not be returned*, i.e., the actual AUC is expected to be lower. The reason is that our estimate is the best out of 1000 tries [86, 34]. The phenomenon is conceptually similar to the multiple hypothesis testing problem in statistics. In small sample sizes, the over-optimism is particularly striking. Recent work shows that most AutoML tools do not correct for this optimism [90]. In this respect, we apply the Bootstrap Bias Correction (**BBC**, hereafter) to our CV estimates [87] that corrects for this optimism. *This leads to returning conservative estimates of performance on average.* As a final report, PROTEUS outputs three objects:

(1) A **surrogate model** of lower-dimensionality for classifying training or new samples regarding their anomalousness. The model approximates the detector model and could be used in its place to visualize, inspect, and interpret.

(2) An **explanation** of anomalies in the form of few selected features that can be manually inspected (e.g., through visualization) to verify the detector's decisions and determine root causes.

(3) An accurate, out-of-sample **estimate of performance**. It can be used to judge the quality of approximation and agreement between the detector and the surrogate model.

Chapter 8

Experimental Evaluation

PROTEUS was implemented in Python 3.6 and evaluated on several synthetic and real-world datasets as described below. The code and the datasets used in our experiments are available in our GitHub repository[1]. All experiments were performed in a Linux Desktop computer with a 4-core Intel i5 processor and 32GB of memory.

8.1 Synthetic and Real Datasets

We focus on datasets where the samples are *independent and identically distributed (i.i.d.)* and contain numerical features. We employ a *synthetic* dataset, where anomalies have been simulated so that a minimal, global, predictive explanation (feature subset) is both achievable and known. The presence of this gold-standard allows us to evaluate how well PROTEUS identifies it. Specifically, we selected randomly one of the 100-dimensional datasets introduced in [37]. Some anomalies have been generated in a way that makes them outliers according to a subset of 2 of these features, call it S_{2d}, and some according to a subset with 3 (other) features, call it S_{3d}. Thus, the subset of these 5 features $S = S_{2d} \cup S_{3d}$ forms the *gold-standard of global explanation for all anomalies.* On this *parent* synthetic dataset, we added irrelevant features with randomly selected values following a normal distribution with zero mean and standard deviation of one. We ended up with 5 synthetic datasets having 20, 40, 60, 80 and 100 dimensions. All of them contain 867 samples with 10 anomalies i.e., the anomaly ratio is $\approx 1\%$. Such datasets have been frequently used in the literature of anomaly explanation [56, 14, 39, 61], because: (a) the features in an explaining subspace (e.g, S_{2d}) are correlated so feature cannot be selected independently; (b) anomalies are recognized as such either in S_{2d} or S_{3d}, but in no other strict subset. Thus, only multivariate detection algorithms and corresponding models will achieve high performance. Hence, PROTEUS must approximate a potentially more complex model.

[1] github link to be added

Dat. Name	#F	#S	A.R.	**IF**	**LOF**	**LODA**
P. Synthetic	5	867	1%	0.96	1.0	0.92
W. Br. Cancer	30	377	5%	0.95	0.94	0.96
Ionosphere	33	358	36%	0.85	0.93	0.87
Arrhythmia	257	452	15%	0.80	0.74	0.75

Table 8.1: Characteristics of datasets and AUC performance of detectors during training. We denote the parent synthetic dataset as P. Synthetic, the number of features and samples as #F and #S and the anomaly ratio as A.R.

We additionally consider *real-world datasets* that are widely-used in the evaluation of anomaly detectors. Specifically, we selected the Wisconsin-Breast Cancer, Ionosphere and Arrhythmia, originally from the UCI Machine Learning repository, as defined for anomaly detection purposes in Outlier Detection DataSets (ODDS) repository[2]. They were chosen to ensure that the detectors employed achieve reasonable performance, and thus explanation makes sense. The dataset characteristics and detector performances are shown in Table 8.1. Wisconsin-Breast Cancer and Ionosphere contain two classes. The minority classes in both datasets are considered as anomalies. For Arrhythmia, eight sub-classes were merged to form the anomaly class. Finally, we added irrelevant features following the procedure described in synthetic datasets constructing three additional datasets per real-world dataset with 30%, 60% and 90% irrelevant feature ratio.

8.2 Experimental Setting

In our experiments, we selected three widely-used unsupervised anomaly detectors that employ different anomalousness criteria, namely Local Outlier Factor (LOF) [10] as a representative of *density-based*, Isolation Forest (IF) [49] as a representative of *isolation-based* and Lightweight On-line Detector of Anomalies (LODA) [65] as a representative of *projection-based* detectors. Regarding the hyper-parameters, for IF we used 100 trees and 256 sub-sample size, for LOF we used $K = 15$ and for LODA we used 100 projection vectors. To assess the predictive power of a surrogate model produced by PROTEUS we stratified and splitted each dataset into 70% for training and 30% was held out for testing. In each dataset, the detectors run on training and test set before adding irrelevant features. The anomaly threshold t is set as the anomaly ratio for each dataset. The detectors performances are demonstrated in Table 8.1.

[2] http://odds.cs.stonybrook.edu/

8.3 Feature Importance Alternatives

We compare the original PROTEUS system, employing general-purpose feature selection methods (call it PROTEUS$_{fs}$), with the PROTEUS pipeline instantiated only with feature importance methods from related explanation methods. We note that these alternatives have been developed to provide *descriptive* explanations; within the PROTEUS pipeline, they are coupled with a classification model, hyper-parameter values are optimized, and they are turned into predictive explanations.

The research question to study is whether *methods specifically developed for explanations in the form of feature importance scores offer additional advantages over the general-purpose methods*, everything else being equal (i.e,. the rest of the PROTEUS pipeline). All alternative methods produce *local* explanations, i.e., for individual samples. Importance scores for a given feature are calculated for each sample (local scores). We compute the local scores only for the anomalous samples. To incorporate them into PROTEUS and select features for global explanations, the local scores are averaged out for each feature to produce a final feature importance score, as proposed in [50]. As a final feature selection, we select the top-K features with the highest importance scores. In our experiments, K is set to 10, which is the maximum number of features allowed to be selected by PROTEUS$_{fs}$ and the feature importance methods. Regarding the hyper-parameters for the feature importance alternatives, we used the ones proposed by the respective authors. We evaluate the following alternatives:

(1) Lightweight On-line Detector of Anomalies or **LODA**, hereafter, [65] is an anomaly detector that also returns local feature importance scores. LODA is included as it has shown an excellent trade-off between computational efficiency and anomaly detection performance as a detector [54]. As a feature importance method is selected as a **representative of a detector-specific explanation method**. As such, the results of its explanation method are shown only for the experiments where LODA is also used as the detector. We should stress that when comparing with LODA, the objective is to approximate its performance as the explanation is strongly coupled to the detection process. The resulting PROTEUS variant is called PROTEUS$_{LODA}$.

(2) Kernel **SHAP** (stands for SHapley Additive exPlanations) [51] is a model-agnostic method for local explanation of predictive models producing local feature scores. It is considered state-of-the-art, having outperformed LIME [67]. As Kernel SHAP does not produce a predictive model itself we consider it as a descriptive method. We use the proposed kernel as in the original publication of SHAP. Kernel SHAP is included as a representative of a **model-agnostic feature importance** method, leading to the variant PROTEUS$_{SHAP}$.

(3) **CA-Lasso** [56], is a representative of a **model-agnostic, local feature importance specifically pertaining to anomaly explanation**. It selects k-nearest neighbors for an outlier a_i and k other random points. To overcome the class imbalance, the authors oversample a_i adding pseudo-samples around it, labelling them as anomalies by assumption, until the two classes are balanced. The

explanation problem is then turned into binary classification solved with Lasso. The feature importance of each feature for a_i corresponds to the Lasso coefficients. It is worth noting the similarities with the general PROTEUS approach: the anomaly explanation problem is turned into binary classification using a classifier that can also produce feature importance scores. The main differences are that (a) PROTEUS directly addresses the global explanation problem, (b) oversampling is supervised (by the detector), (c) numerous algorithms and hyper-parameter values are searched, (d) the out-of-sample (predictive) performance is estimated. The resulting PROTEUS variant is called PROTEUS$_{CA-Lasso}$.

8.3.1 PROTEUS Performance Estimation

The objective of this experiment is to assess the effect of PROTEUS design choices, specifically the BBC and Grouping, to provide an accurate performance estimation. Figure 8.1 depicts the train estimates and test performance when PROTEUS is employed with the design choices described in Section 7, i.e., BBC and CV with Grouping. The dashed black diagonal line indicates the zero bias: points above the diagonal indicate underestimation (negative bias) and below overestimation (optimistic bias). To show the accuracy of the estimation of PROTEUS design choices, we fit a loess curve[3] on train and test performances for every combination (258 in total) of datasets (synthetic and real), detectors (IF, LOF and LODA) and feature selection methods (general purpose and feature importance methods). Ideally, we would want the loess curve to fit exactly the diagonal. Observe that with lower AUC performances PROTEUS tends to overestimate while with higher performances PROTEUS returns a more conservative estimation. In both cases, the points are close to the ideal diagonal line.

To further show the efficacy of the proposed design choices to provide an accurate performance estimation, in Figure 8.2 we compare the loess curves for train and test estimates for (i) BBC and Grouping (our design choices), (ii) no BBC (i.e., CV estimate) and Grouping (iii) BBC and no Grouping and (iv) no BBC and no Grouping. To quantify the bias for each of the four alternatives, we use the Residual Sum of Squares (RSS) to measure the discrepancy between the train and test performance. When PROTEUS is employed with BBC and Grouping (i), it gives the most accurate estimation of out-of-sample performance (with $RSS_{(i)}$ = 0.05) than when using any of the three alternative design choices (with $RSS_{(ii)}$ = 0.88, $RSS_{(iii)}$ = 0.11 and $RSS_{(iv)}$ = 0.25).

8.4 Relevant Features Identification Accuracy

The goal of this experiment is to verify whether the features discovered during the training phase by PROTEUS$_{fs}$ and the feature importance alternatives are part of the gold-standard feature subset S. For this experiment we used the *synthetic*

[3] https://en.wikipedia.org/wiki/Local_regression

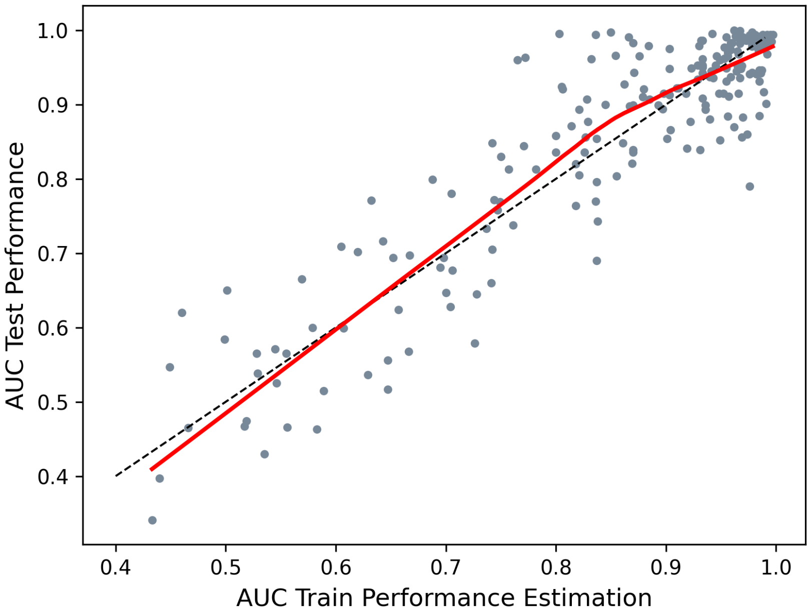


Figure 8.1: Bias between train and test AUC performances as reported by PROTEUS implemented with BBC and CV with grouping. Each point represents the performance for a particular pipeline, e.g. when PROTEUS explains LOF in Arrhythmia using the general-purpose feature selection methods. A more general bias trend is captured by fitting a loess curve to the obtained performances

datasets. To assess the quality of the global explanation E in terms of features, we compute $precision(S, E) = \frac{|S \cap E|}{|E|}$ and $recall(S, E) = \frac{|S \cap E|}{|S|}$. As we select the top-10 features to form the explanation and S contains 5 features, the *precision* for the feature importance alternative methods will be up to 0.5. The recall and precision curves are depicted in Figure 8.3. General-purpose feature selection methods employed by PROTEUS$_{fs}$ exhibit the highest precision never dropping below 0.5, independently of the employed detector or dataset dimensionality. We observed that precision is 0.5 when Lasso is selected and higher when FBED is selected. We should stress that SES was never selected by PROTEUS for the synthetic datasets. FBED removed most of the irrelevant features leading to a predictive model with less than 10 features to approximate the decision boundary of the corresponding detector. PROTEUS$_{fs}$ achieves almost optimal recall regardless of the dimensionality and the employed detector. A slight drop in recall is observed when the precision higher than 0.8 (achieved only by FBED), while recall is optimal when Lasso is selected. Moreover, PROTEUS$_{fs}$ general-purpose methods are robust to increasing data dimensionality and irrelevant feature ratio where CA-Lasso and SHAP seem to be particularly sensitive.

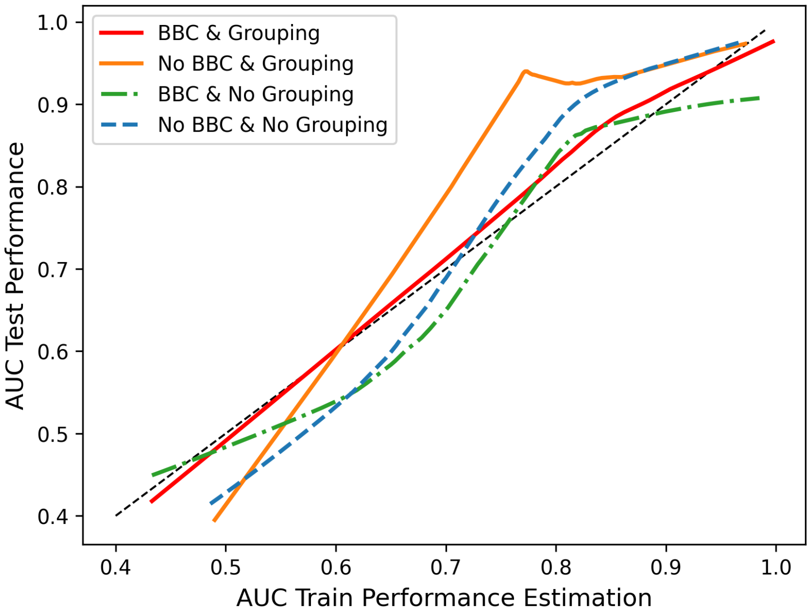


Figure 8.2: Bias between train and test AUC performance as reported by PROTEUS implemented with 4 alternatives

8.5 PROTEUS Generalization Performance

The objective of this experiment is to assess the generalization performance of PROTEUS without (PROTEUS$_{full}$) and with feature selection (PROTEUS$_{fs}$) as well as with the various feature importance alternatives, (PROTEUS$_{CA-Lasso}$, PROTEUS$_{SHAP}$, PROTEUS$_{LODA}$). Figure 8.4 depicts the AUC performance for each method in test set. Regarding the synthetic datasets, PROTEUS$_{fs}$ achieves very high AUC across the increasing data dimensionality with a minimum of 0.96. CA-Lasso and SHAP instead exhibit lower performances as they do not retrieve, as showed in the previous experiment, many of the relevant features. Observe that in the synthetic dataset PROTEUS$_{fs}$ generalizes better than PROTEUS$_{full}$, i.e., when using all the available features.

Regarding the real datasets, similar trends are observed with PROTEUS$_{fs}$ achieving consistently a very high generalization performance with a minimum of 0.8 in Arrhythmia in the presence of 2,570 dimensions and 90% irrelevant feature ratio. PROTEUS$_{fs}$ seems to approximate in a detector-agnostic manner, the optimal performance of LODA's feature importance method when LODA is used as the detection algorithm. This is due to the fact that LODA's explanations are tailored to its detection algorithm; however, if LODA's detection performance was poor in a dataset, the provided explanation would be of less value for the analysts. Moreover, Figure 8.5 demonstrates the effect of the proposed oversampling

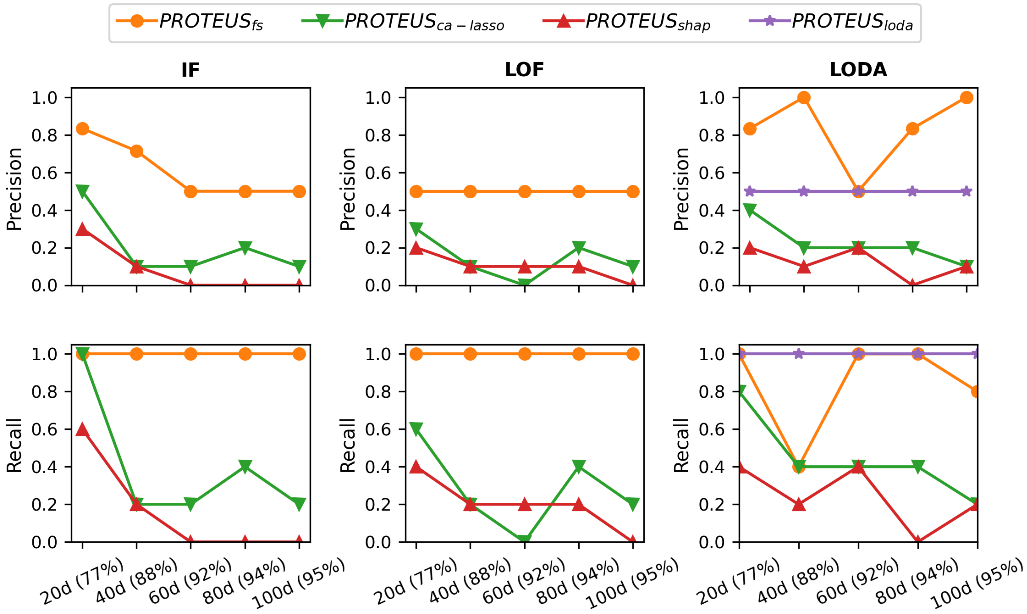


Figure 8.3: Precision and Recall performance of discovered features when explaining IF, LOF and LODA on synthetic datasets w.r.t. increasing data dimensionality (irrelevant feature ratio)

approach on generalization performance of PROTEUS$_{fs}$ w.r.t. the increasing number of pseudo-samples per anomaly for the real datasets. For Breast Cancer the oversampling does not increase the performance due to the "ceiling effect". For Ionosphere and Arrhythmia we observe that the performance is increased when explaining LODA.

In a nutshell, the general-purpose feature selection methods employed by PROTEUS$_{fs}$, are able to discover the relevant features leading to predictive models with very high performance regardless of the data dimensionality (and the increasing relevant feature ratio) and capture accurately the decision boundary of every employed unsupervised detector. A scatter plot of the Wisconsin-Breast Cancer dataset over two 2-dimensional explanations produced by PROTEUS$_{fs}$ for LODA's and LOF's decision boundary are demonstrated in Figures 8.6a and 8.6b respectively.

8.6 PROTEUS Efficiency

In this experiment, we demonstrate the execution time of PROTEUS pipeline as well as its general-purpose feature selection methods. In Figure 8.7, we compare the runtime of PROTEUS AutoML pipeline to produce the predictive explanation including all the procedures involved in Figure 7.1 with the unsupervised summarization anomaly explanation algorithms evaluated in Part I of our work. Note that the comparisons were performed for $5d$ explanations which was the upper

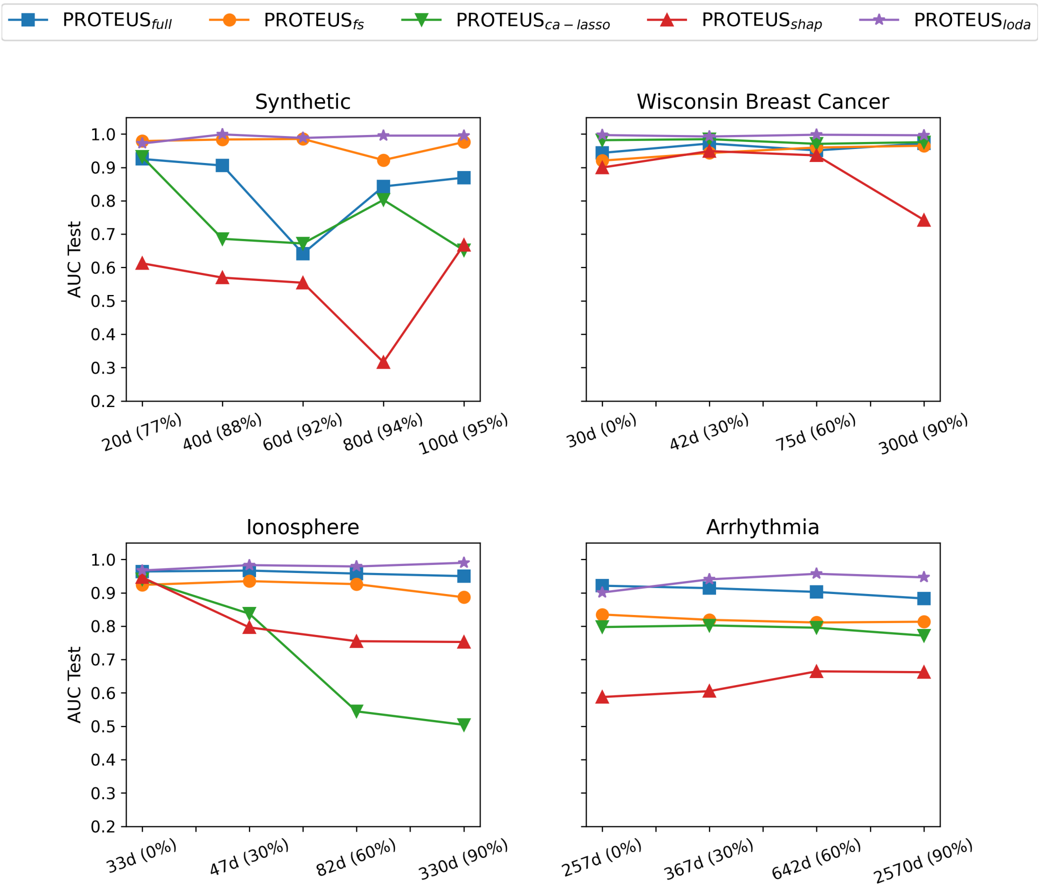


Figure 8.4: AUC test performance averaged over the three detectors on synthetic and real datasets w.r.t. increasing dimensionality (irrelevant feature ratio)

limit of our comparative evaluation of Part I. Unlike all other explanation methods, PROTEUS exhibits an almost constant execution time as the dimensionality of synthetic data increases. However, PROTEUS is three times slower on average than HiCS. On the contrary, PROTEUS is significantly faster than LookOut that performs an exhaustive search, especially in higher data and explanation dimensionality. The major difference between a descriptive and an AutoML explainer such as PROTEUS is that the former's execution time is highly tailored to the data and explanation dimensionality, while the latter depends mainly on the number of configurations used to find the most suitable pipeline for a given dataset. To conclude, the major advantage of PROTEUS stems from the fact there is no need to recompute the explanation for every new batch of data, avoiding that way additional computational costs.

Figure 8.8 depicts the runtime comparison between the general-purpose feature selection algorithms employed by PROTEUS and the ad-hoc feature importance methods. The general-purpose feature selection algorithms require less than two

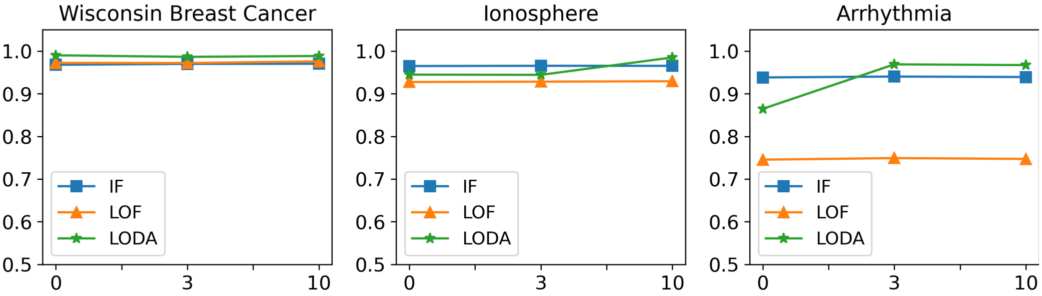
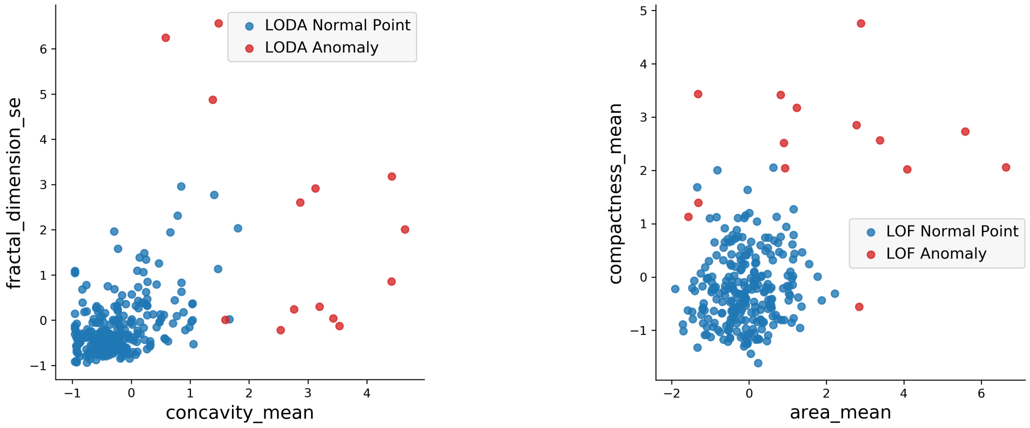


Figure 8.5: Effect of increasing pseudo-sample size per anomaly on AUC test performance



(a)

(b)

Figure 8.6: Subspaces Explaining Anomalies in Cancer Data

seconds on average in 100-dimensions to select features, exhibiting a steady execution time, while SHAP's cost is particularly sensitive as data dimensionality increases.

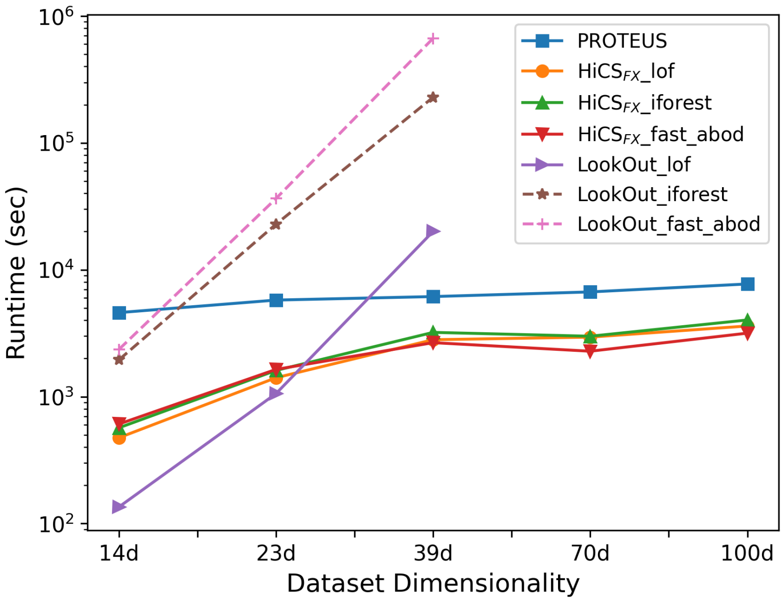


Figure 8.7: Total execution time of PROTEUS, HiCS and LookOut on synthetic datasets

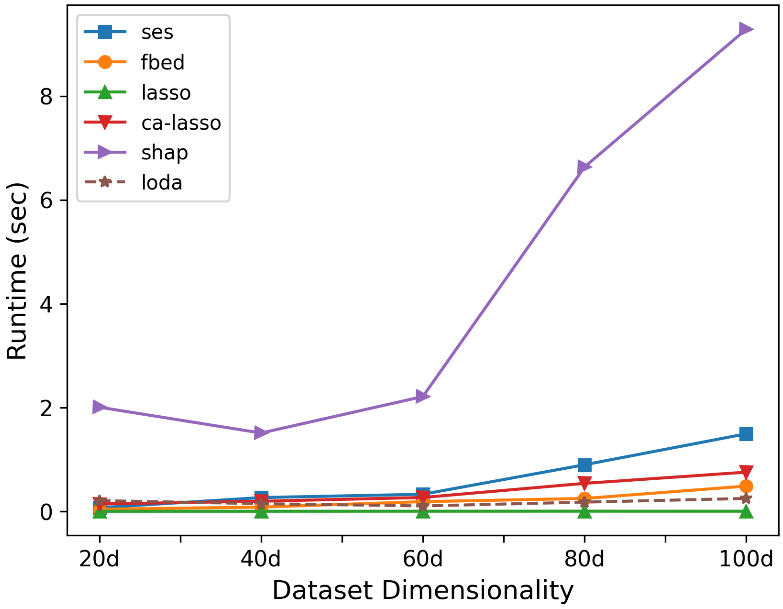


Figure 8.8: Average execution time of feature selection and importance methods on synthetic datasets

Chapter 9

Conclusion and Future Work

In this thesis, we focused on the explanation of data anomalies. In the first part, we demonstrated a thorough evaluation of unsupervised anomaly explainers addressing several missing insights regarding the performance of existing anomaly explanation and summarization algorithms under realistic settings. We underlined the main challenge that stems from the lack of inherent pruning properties to effectively search the exponential space. Existing subspace search strategies exploit the distributional characteristics either: (i) of data such as features correlation in subspaces (HiCS [36]) or (ii) of scores given by an anomaly detector in subspaces (LookOut [29], Beam [62] and RefOut [38]). The former strategy is effective when highly clustered anomalies over correlated features are contained in datasets regardless of their dimensionality, while the latter is effective in low explanation dimensionality where the anomaly detectors can discriminate accurately the anomalies from the inliers. It remains open to assess whether the low dimensional subspaces retrieved by an explainer are projections of a high dimensional subspace fully explaining a specific point.

We should additionally note that the detection of anomalies in LOF, ABOD and iForest, is actually decoupled from the search of subspaces likely to contain them. HiCS, RefOut and Beam instead are *explaining anomaly detectors* that rely on per-subspace measures to quantify the explanation quality of subspaces. We are planning to extend our testbed with recent works [83] taking into account the relationship between subspaces using a dimension-based measure of their explanation quality. Moreover, in case of recurring anomaly patterns, it is also interesting to benchmark group-based explanation summarization techniques [53]. Another interesting aspect would be to investigate anomaly explanation in stream processing settings such as LODA [65].

At the second part of this thesis, we proposed the first methodology for producing predictive, global anomaly explanations in a detector-agnostic fashion. In particular, we show how with adequate design choices regarding rare class oversampling as well as unbiased performance estimation of ML pipelines, generating predictive, global anomaly explanations boils down to an AutoML problem. As yielded from our experiments, PROTEUS is not only able to discover explaining subspaces of features relevant to anomalies, but it can also construct predictive models that approximate effectively and robustly the decision boundary of popular unsupervised detectors (e.g., IF, LOF, LODA).

As future work, we plan to make PROTEUS more efficient by leveraging automatic approaches that can optimize the performance of any given learning algorithm to the problem at hand [74]. Another computational cutoff to consider is the Bootstrap Bias Corrected with Dropping Cross-Validation (BBCD-CV) protocol [87]. This protocol can lead to substantial computational savings as numerous configurations can be dropped after just a few folds before completing the full K-fold CV on them. Moreover, it would be interesting to approximate the decision boundary of a detector directly from the provided anomaly scores rather than converting them to binary labels. Hence, one could transform the explanation problem into regression with feature selection.

Chapter 10

Acknowledgments

This work was partially supported by SAP company (France, Paris 92309 Levallois Perret, +33 1 46 17 70 00) and by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant” (Project Number: 1941).

Bibliography

[1] Firas Abuzaid, Peter Bailis, Jialin Ding, Edward Gan, Samuel Madden, Deepak Narayanan, Kexin Rong, and Suri Sahaana. Macrobase: Prioritizing attention in fast data. *ACM Trans. Database Syst.*, 43(4):15:1–15:45, December 2018.

[2] Herman Aguinis, Ryan K. Gottfredson, and Harry Joo. Best-practice recommendations for defining, identifying, and handling outliers. *Organizational Research Methods*, 16(2):270–301, jan 2013.

[3] Fabrizio Angiulli, Fabio Fassetti, and Luigi Palopoli. Detecting outlying properties of exceptional objects. *ACM Trans. Database Syst.*, 34(1):7:1–7:62, 2009.

[4] Fabrizio Angiulli, Fabio Fassetti, and Luigi Palopoli. Discovering characterizations of the behavior of anomalous subpopulations. *IEEE Trans. Knowl. Data Eng.*, 25(6):1280–1292, 2013.

[5] Fabrizio Angiulli, Fabio Fassetti, Luigi Palopoli, and Giuseppe Manco. Outlying property detection with numerical attributes. *CoRR*, abs/1306.3558, 2013.

[6] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831, 2010.

[7] Aline Bessa, Juliana Freire, Tamraparni Dasu, and Divesh Srivastava. Effective discovery of meaningful outlier relationships. *ACM/IMS Trans. Data Sci.*, 1(2), June 2020.

[8] Lucien Birge and Yves Rozenholc. How many bins should be put in a regular histogram. *ESAIM: Probability and Statistics*, 10:24–45, 2006.

[9] Klemens Böhm, Fabian Keller, Emmanuel Müller, Hoang Vu Nguyen, and Jilles Vreeken. CMI: an information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA*, pages 198–206. SIAM, 2013.

[10] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *SIGMOD Conference*, 2000.

[11] Guilherme O. Campos, Arthur Zimek, Jorg Sander, Ricardo J. Campello, Barbora Micenkova, Erich Schubert, Ira Assent, and Michael E. Houle. On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Min. Knowl. Discov.*, 30(4):891–927, July 2016.

[12] Lei Cao, Qingyang Wang, and Elke A. Rundensteiner. Interactive outlier exploration in big data streams. *Proc. VLDB Endow.*, 7(13):1621–1624, August 2014.

[13] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

[14] Xuan-Hong Dang, Ira Assent, Raymond T. Ng, Arthur Zimek, and Erich Schubert. Discriminative features for identifying and interpreting outliers. In *ICDE*, pages 88–99, 2014.

[15] Xuan-Hong Dang, Barbora Micenková, Ira Assent, and Raymond T. Ng. Local outlier detection with interpretation. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Zelezný, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III*, volume 8190 of *Lecture Notes in Computer Science*, pages 304–320. Springer, 2013.

[16] Tamraparni Dasu, Ji Meng Loh, and Divesh Srivastava. Empirical glitch explanations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 572–581. ACM, 2014.

[17] Manuel Fernández Delgado, Eva Cernadas, Senén Barro, and Dinani Gomes Amorim. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.*, 15(1):3133–3181, 2014.

[18] Remi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, 74:406–421, 2018.

[19] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[20] Lei Duan, Guanting Tang, Jian Pei, James Bailey, Akiko Campbell, and Changjie Tang. Mining outlying aspects on numeric data. *Data Mining and Knowledge Discovery*, 29:1116–1151, 2014.

[21] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE International Conference on Computer*

Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 3449–3457. IEEE Computer Society, 2017.

[22] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. *Pattern Recogn. Lett.*, 31(14):2225–2236, October 2010.

[23] Michail Giannoulis. Benchmarking Anomaly Detectors on Streaming Data. Master's thesis, Computer Science Department, University of Crete, Voutes-Heraklion, 2020.

[24] Ioana Giurgiu and Anika Schumann. Additive explanations for anomalies detected from multivariate temporal data. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2245–2248, New York, NY, USA, 2019. Association for Computing Machinery.

[25] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS One*, 11(4), 4 2016.

[26] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS One*, 11(4), 4 2016.

[27] Xiaoyi Gu, Leman Akoglu, and Alessandro Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. In *NeurIPS*, pages 10921–10931, 2019.

[28] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2712–2721. JMLR.org, 2016.

[29] Nikhil Gupta, Dhivya Eswaran, Neil Shah, Leman Akoglu, and Christos Faloutsos. Beyond outlier detection: Lookout for pictorial explanation. In *ECML/PKDD*, 2018.

[30] Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *ICIC*, 2005.

[31] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[32] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *TKDE*, 21:1263–1284, 2009.

[33] Chih-Wei Hsu, Chih-Chung Chang, and C. Lin. A practical guide to support vector classification. *BJU International*, 2008.

[34] David D. Jensen and Paul R. Cohen. Multiple comparisons in induction algorithms. *do. Learn.*, 38(3):309–338, 2000.

[35] Vimalkumar Jeyakumar, Omid Madani, Ali Parandeh, Ashutosh Kulshreshtha, Weifei Zeng, and Navindra Yadav. Explainit! – a declarative root-cause analysis engine for time series data. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, pages 333–348. ACM, 2019.

[36] Fabian Keller, Emmanuel Müller, and Klemens Böhm. Hics: High contrast subspaces for density-based outlier ranking. *2012 IEEE 28th International Conference on Data Engineering*, pages 1037–1048, 2012.

[37] Fabian Keller, Emmanuel Müller, and Klemens Böhm. Hics: High contrast subspaces for density-based outlier ranking. In *ICDE*, pages 1037–1048, 2012.

[38] Fabian Keller, Emmanuel Müller, Andreas Wixler, and Klemens Böhm. Flexible and adaptive subspace search for outlier analysis. In *CIKM*, 2013.

[39] Fabian Keller, Emmanuel Müller, Andreas Wixler, and Klemens Böhm. Flexible and adaptive subspace search for outlier analysis. In *CIKM*, pages 1381–1390, 2013.

[40] Edwin M. Knorr and Raymond T. Ng. Finding intensional knowledge of distance-based outliers. In Malcolm P. Atkinson, Maria E. Orlowska, Patrick Valduriez, Stanley B. Zdonik, and Michael L. Brodie, editors, *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 211–222. Morgan Kaufmann, 1999.

[41] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 2017.

[42] Maria Kontaki, Anastasios Gounaris, Apostolos N. Papadopoulos, Kostas Tsichlas, and Yannis Manolopoulos. Efficient and flexible algorithms for monitoring distance-based outliers over data streams. *Inf. Syst.*, 55:37–53, 2016.

[43] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In Thanaruk

Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu Bao Ho, editors, *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009, Bangkok, Thailand, April 27-30, 2009, Proceedings*, volume 5476 of *Lecture Notes in Computer Science*, pages 831–838. Springer, 2009.

[44] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *PAKDD*, 2009.

[45] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 444–452. ACM, 2008.

[46] Chia-Tung Kuo and Ian Davidson. A framework for outlier description using constraint programming. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1237–1243. AAAI Press, 2016.

[47] Vincenzo Lagani, Giorgos Athineou, Alessio Farcomeni, Michail Tsagris, Ioannis Tsamardinos, et al. Feature selection with the r package mxm: Discovering statistically equivalent feature subsets. *Journal of Statistical Software*, 2017.

[48] A. Lavin and S. Ahmad. Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 38–44, Dec 2015.

[49] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *ICDM*, pages 413–422, 2008.

[50] Scott M. Lundberg, Gabriel G. Erion, Hugh Chen, Alex J. DeGrave, Jordan M Prutkin, Bala G. Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2:56–67, 2020.

[51] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, pages 4765–4774, 2017.

[52] Meghanath Macha and Leman Akoglu. Explaining anomalies in groups with characterizing subspace rules. *Data Min. Knowl. Discov.*, 32(5):1444–1480, 2018.

[53] Meghanath Macha and Leman Akoglu. Explaining anomalies in groups with characterizing subspace rules. *Data Min. Knowl. Discov.*, 32(5):1444–1480, September 2018.

[54] Emaad A. Manzoor, Hemank Lamba, and Leman Akoglu. xstream: Outlier detection in feature-evolving data streams. In Yike Guo and Faisal Farooq, editors, *KDD*, pages 1963–1972, 2018.

[55] David Mease, Abraham J. Wyner, and Andreas Buja. Boosted classification trees and class probability/quantile estimation. *J. Mach. Learn. Res.*, 8:409–439, 2007.

[56] Barbora Micenková, Raymond T. Ng, Xuan-Hong Dang, and Ira Assent. Explaining outliers by subspace separability. In *ICDM*, pages 518–527, 2013.

[57] Christoph Molnar. *Interpretable Machine Learning*. 2019. `https://christophm.github.io/interpretable-ml-book/`.

[58] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.*, 73:1–15, 2018.

[59] George L. Nemhauser and Laurence A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Math. Oper. Res.*, 3:177–188, 1978.

[60] Hien M. Nguyen, Eric W. Cooper, and Katsuari Kamei. Borderline oversampling for imbalanced data classification. *Int. J. Knowl. Eng. Soft Data Paradigms*, 3:4–21, 2011.

[61] Xuan Vinh Nguyen, Jeffrey Chan, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, and Jian Pei. Scalable outlying-inlying aspects discovery via feature ranking. In *PAKDD*, pages 422–434, 2015.

[62] Xuan Vinh Nguyen, Jeffrey Chan, Simone Romano, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, and Jian Pei. Discovering outlying aspects in large datasets. *Data Mining and Knowledge Discovery*, 30:1520–1555, 2016.

[63] Abdul Nurunnabi and Geoff West. Outlier detection in logistic regression: A quest for reliable knowledge from predictive modeling and classification. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 643–652, Dec 2012.

[64] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jacob VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[65] Tomás Pevný. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102:275–304, 2015.

[66] Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. Data lifecycle challenges in production machine learning: A survey. *SIGMOD Rec.*, 47(2):17–28, December 2018.

[67] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *KDD*, pages 1135–1144, 2016.

[68] Marko Robnik-Sikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20:589–600, 2008.

[69] Sudeepa Roy, Laurel Orr, and Dan Suciu. Explaining query answers with explanation-ready databases. *Proc. VLDB Endow.*, 9(4):348–359, December 2015.

[70] Saket Sathe and Charu C. Aggarwal. Subspace outlier detection in linear time with randomized hashing. *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 459–468, 2016.

[71] Erich Schubert and Arthur Zimek. Elki: A large open-source library for data analysis - elki release 0.7.5 "heidelberg". *ArXiv*, abs/1902.03616, 2019.

[72] Md Amran Siddiqui, Alan Fern, Thomas G. Dietterich, and Weng-Keen Wong. Sequential feature explanations for anomaly detection. *ACM Trans. Knowl. Discov. Data*, 13(1):1:1–1:22, 2019.

[73] Md Amran Siddiqui, Alan Fern, Thomas G. Dietterich, and Weng-Keen Wong. Sequential feature explanations for anomaly detection. *ACM Trans. Knowl. Discov. Data*, 13(1):1:1–1:22, January 2019.

[74] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2960–2968, 2012.

[75] Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18, 2010.

[76] Erik Strumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, 41(3):647–665, 2014.

[77] Wanhua Su, Yan Yuan, and Mu Zhu. A relationship between the average precision and the area under the roc curve. In *ICTIR '15*, 2015.

[78] Sharmila Subramaniam, Themis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*, pages 187–198. ACM, 2006.

[79] Ayman Taha and Ali S. Hadi. Anomaly detection methods for categorical data: A review. *ACM Comput. Surv.*, 52(2), May 2019.

[80] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, pages 267–288, 1996.

[81] Kai Ming Ting, Takashi Washio, Jonathan R. Wells, and Sunil Aryal. Defying the gravity of learning curve: a characteristic of nearest neighbour anomaly detectors. *Machine Learning*, 106:55–91, 2016.

[82] Luan Tran, Liyue Fan, and Cyrus Shahabi. Distance-based outlier detection in data streams. *Proc. VLDB Endow.*, 9(12):1089–1100, August 2016.

[83] Holger Trittenbach and Klemens Böhm. Dimension-based subspace search for outlier detection. *International Journal of Data Science and Analytics*, 7(2):87–101, Mar 2019.

[84] Ioannis Tsamardinos and Constantin F. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, AISTATS 2003*, 2003.

[85] Ioannis Tsamardinos, Giorgos Borboudakis, Pavlos Katsogridakis, Polyvios Pratikakis, and Vassilis Christophides. A greedy feature selection algorithm for big data of high dimensionality. *Machine Learning*, 108(2):149–202, 2019.

[86] Ioannis Tsamardinos, Giorgos Borboudakis, Pavlos Katsogridakis, Polyvios Pratikakis, and Vassilis Christophides. A greedy feature selection algorithm for big data of high dimensionality. *Mach. Learn.*, 108(2):149–202, 2019.

[87] Ioannis Tsamardinos, Elissavet Greasidou, and Giorgos Borboudakis. Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Mach. Learn.*, 107(12):1895–1922, 2018.

[88] Adam White and Artur S. d'Avila Garcez. Measurable counterfactual local explanations for any classifier. In *ECAI 2020 - 24th European Conference on Artificial Intelligence - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2529–2535, 2020.

[89] Eugene Wu and Samuel Madden. Scorpion: Explaining away outliers in aggregate queries. *Proc. VLDB Endow.*, 6(8):553–564, June 2013.

[90] Iordanis Xanthopoulos. A Qualitative, Quantitative and User-based Methodology of Automated Machine Learning Systems Evaluation. Master's thesis, Computer Science Department, University of Crete, Voutes-Heraklion, 2020.

[91] Hui Xiong, Gaurav Pandey, Michael Steinbach, and Vipin Kumar. Enhancing data analysis with noise removal. *IEEE Trans. on Knowl. and Data Eng.*, 18(3):304–319, March 2006.

[92] Jiawei Yang, Susanto Rahardja, and Pasi Fränti. Outlier detection: how to threshold outlier scores? In *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing, AIIPCC 2019*, pages 37:1–37:6, 2019.

[93] Haopeng Zhang, Yanlei Diao, and Alexandra Meliou. Exstream: Explaining anomalies in event stream monitoring. In *20th International Conference on Extending Database Technology (EDBT)*, pages 156–167, March 2017.

[94] Haopeng Zhang, Yanlei Diao, and Alexandra Meliou. Exstream: Explaining anomalies in event stream monitoring. In Volker Markl, Salvatore Orlando, Bernhard Mitschang, Periklis Andritsos, Kai-Uwe Sattler, and Sebastian Breß, editors, *EDBT*, pages 156–167, 2017.

[95] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *J. Mach. Learn. Res.*, 20:96:1–96:7, 2019.