A Comparative Evaluation of Anomaly Explanation Algorithms

Nikolaos Myrtakis*
University of Crete
Heraklion, Greece
myrtakis@csd.uoc.gr

Vassilis Christophides†
ENSEA, ETIS
Cergy, France
vassilis.christophides@ensea.fr

Eric Simon
SAP, France
Paris, France
eric.simon@sap.com

ABSTRACT

Detection of anomalies (i.e., *outliers*) in multi-dimensional data is a well-studied subject in machine learning. Unfortunately, unsupervised detectors provide no explanation about why a data point was considered as abnormal or which of its features (i.e. subspaces) exhibit at best its outlyingness. Such *outlier explanations* are crucial to diagnose the root cause of data anomalies and enable corrective actions to prevent or remedy their effect in downstream data processing. In this work, we present a comprehensive framework for comparing different unsupervised outlier explanation algorithms that are domain and detector-agnostic.

Using real and synthetic datasets, we assess the effectiveness and efficiency of two *point explanation* algorithms (Beam [28] and RefOut [18]) ranking subspaces that best explain the outlyingness of *individual* data points and two *explanation summarization* algorithms (LookOut [15] and HiCS [17]) ranking subspaces that best exhibit as *many outlier points* from inliers as possible. To the best of our knowledge, this is the first detailed evaluation of existing explanation algorithms 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?

1 INTRODUCTION

Detecting and diagnosing data anomalies are important tasks in data processing pipelines used to build industrial-strength Machine Learning (ML) systems [32]. Clearly, data points that significantly deviate from other points in a dataset may be systematic errors, i.e., *outliers*, or may manifest changes in the data generation process per se, i.e., *novelties*, that decrease the accuracy of the predictive models constructed downstream [29, 48]. In scientific and industrial monitoring applications, anomaly detection is often the ultimate goal of the data analysis as it enables the identification of unusual measurements (e.g., related to faults, bio-indices, etc.) and/or of suspicious activities (e.g. intrusions, fraud, etc.). Several unsupervised algorithms for anomaly detection have been proposed [2, 51] using different methods (e.g., proximity or isolation based) to distinguish outliers from inliers when labels of data points are impossible or difficult to obtain. Unfortunately, these algorithms do not explain why a data point was considered as abnormal, leaving analysts with no guidance about where to begin their investigation.

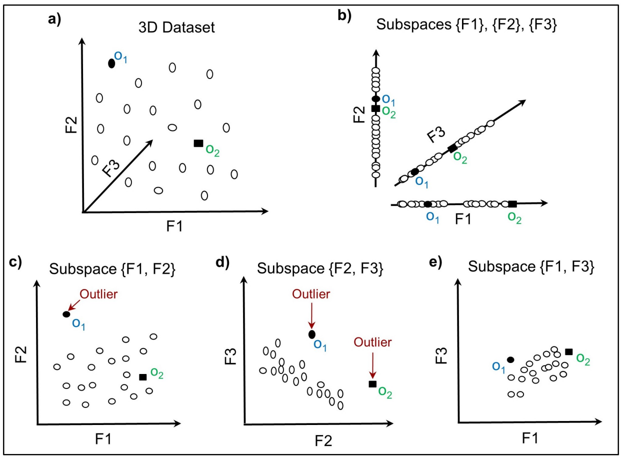


Figure 1: A 3d dataset with three 1d and 2d feature subspaces

In this paper, we focus on algorithms explaining the *outlyingness* aspects of multi-dimensional data points in the form of subspaces of data features that best explain why a given outlier deviates the most from the inliers. Such *explanations* are crucial to diagnose the root cause of data anomalies [3] and enable corrective actions to prevent or remedy their effect in downstream data processing (e.g. by repairing data errors or retraining the predictive models for concept drifts).

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 outlyingness of points $o1$ and $o2$ depicted by a black circle and a black square in Figure 1-a). In the full dimensional space of the dataset, $o1$ exhibits a small deviation from most of the other points in the dataset while $o2$ looks like an inlier although it exhibits a significant outlyingness when considering the subset of features $\{F_2, F_3\}$ (see Figure 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 feature sets (aka subspaces) points exhibit a high outlyingness. None of the 1d subspaces $\{F_1\}$, $\{F_2\}$ and $\{F_3\}$ explain the outlyingness of the two points (see Figure 1-b). The same is true for the 2d subspace $\{F_1, F_3\}$ (see Figure 1-e). Subspace $\{F_1, F_2\}$ explains the outlyingness of $o1$ only (see Figure 1-c), while $\{F_2, F_3\}$ explains the outlyingness of both points (see Figure 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 $o2$.

*Work was done while the author was working at SAP.

†This work received funding by the CY Initiative of Excellence (grant "Investissements d'Avenir" ANR-16-IDEX-0008) and developed during the author stay at the CY Advanced Studies, whose support is gratefully acknowledged.

Series ISSN: 2367-2005

10.5441/002/edbt.2021.10

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). Our choice of explanation algorithms is motivated by the fact that no detector is good in all possible settings w.r.t data characteristics (see the conclusions of several experimental studies [6, 8, 14]). Hence we are interested in decoupling the outlier explanation from detection, in contrast to several techniques proposed in Explainable Artificial Intelligence (XAI) such as output contribution of attribute values [24, 33] or partial dependence plots [11].

We evaluate two *point explanation algorithms*, *RefOut* [18] and *Beam* [28], that rank subspaces best explaining the outlyingness of *individual* data points, and two *explanation summarization algorithms*, *LookOut* [15] and *HICS* [17], that rank subspaces best explaining the outlyingness of as *many* outlier points as possible. These algorithms rely on outlyingness criteria of existing detectors such as Local Outlier Factor (LOF) [5], Angle Based Outlier Detection (ABOD) [21] or Isolation Forest (iForest) [23].

Although there exist several efforts for benchmarking outlier detectors in batch [6, 8, 12, 42] and stream [22, 43] 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 aims to answer the following questions: ***1.*** *Is it effective to combine any explanation algorithm with any off-the-shelf outlier detector?* ***2.*** *How is the behavior of an outlier detection and explanation pipeline affected by the number of features or their correlation in a dataset?* ***3.*** *What is the quality of summaries in the presence of outliers explained by subspaces of different dimensionality?*

The remaining of the paper is organized as follows. Section 2 introduces the outlier detectors and the point explanation and summarization algorithms we integrated in our experimental testbed. Section 3 details the pipelines of algorithms, the datasets as well as the evaluation metric used in our testbed. Section 4 presents the conducted experiments and the conclusions drawn regarding the missing insights. Section 5 surveys additional explanation algorithms for data in rest or in motion and justify why they have not been included in our study. Section 6 concludes our work and presents plans for future research.

2 OUTLIER DETECTION AND EXPLANATION ALGORITHMS

2.1 Unsupervised Outlier Detectors

Several methods have been proposed in the literature to measure the abnormality of a data point in a dataset. In the following, we survey three unsupervised methods that are widely used for detecting outliers in datasets with multiple numerical[1] features [6, 8, 12, 13, 42]. As the objective of outlier explanation is to retrieve subspaces where the outliers are clearly separable from inliers, we did not include any subspace-based outlier detector [20, 36] to assess the quality of a particular subspace examined by the explainer. 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).

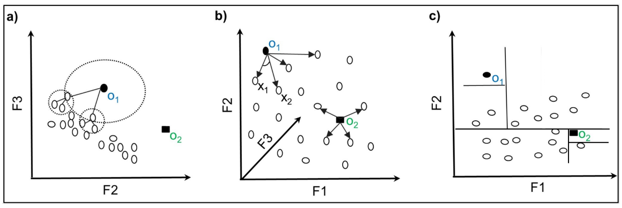


Figure 2: Examples of outliers in different subspaces detected by (a) LOF, (b) Fast ABOD and (c) iForest

Density-Based methods, such as Local Outlier Factor (LOF) [5] take into account the local density of points when searching for outliers. An example of outliers detected by LOF is illustrated in Figure 2-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*:

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

LOF's time complexity is $O(N^2)$, 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.

Angle-Based methods compute for each given point, the angles to other data points N. The Angle Based Outlier Detector (ABOD) [21] uses the variance of these angles as an outlyingness score. For example, as we can see in Figure 2-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:

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

As ABOD's time complexity is $O(N^3)$, we are focusing on an efficient ABOD variant ($O(kN^2)$), called *Fast ABOD*, which computes the angles of a particular point only to its k-nearest neighbors. Small angle variance results to 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.

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-c) the point o_1 is an outlier as it needs less partitions to be isolated compared to the inlier o_2. Isolation Forest

[1] Anomaly detection methods for categorical data [41] are outside the scope of this work.

(iForest) [23] 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 outliers 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 ($O(tnlogn)$) by exploiting subsampling and by eliminating the heavy cost of distance computation. Being agnostic to the distances (or densities) of points, iForest is able to detect outliers effectively even if they are lying on less dense areas than the majority of the points.

2.2 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 [28], 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 [18] 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 3. 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 3, 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 Z-score as follows:

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

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 [46] 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 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.

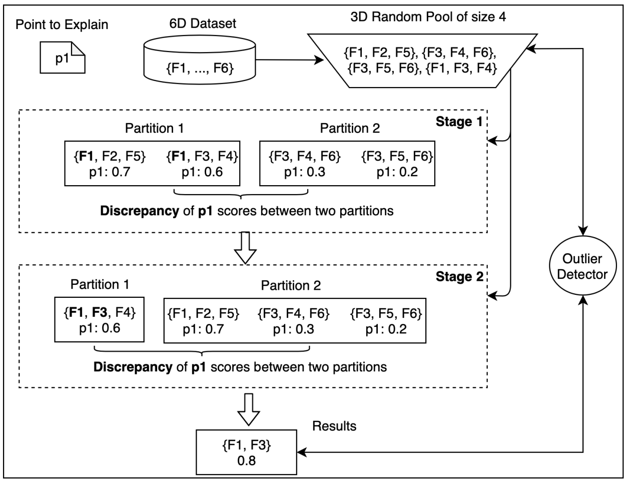


Figure 3: RefOut steps to find $2d$ subspaces from a $6d$ dataset to explain the point p1

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 [28] 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 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. 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

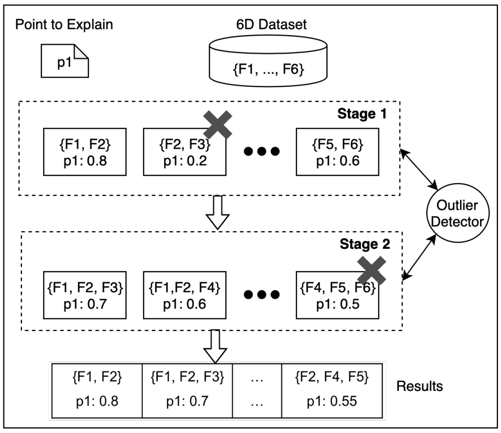
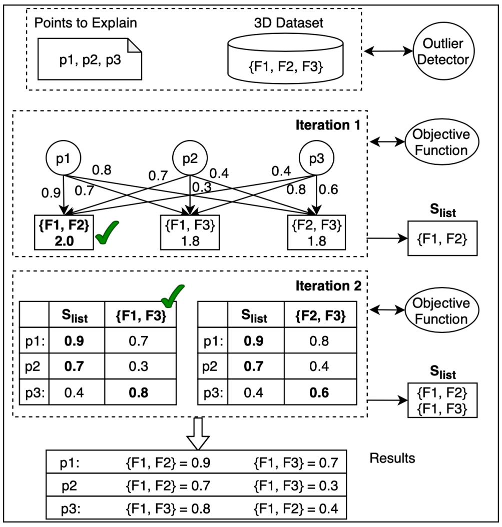


Figure 4: Beam steps to find subspaces up to 3 dimensions from a 6d dataset to explain the point p1



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

	Slist	{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 5: LookOut steps to find 2d subspaces from a 3d dataset with budget b = 2 (bold values indicate the highest scores per table row)

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 $Beam_{FX}$.

2.3 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 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. We should stress that explanation summarization is different from group identification and explanation. In the former case, we consider all the to-be-explained points as one group, while on the latter, the objective is to identify these anomalous groups and retrieve explaining subspaces that segregate each group from the normal instances [25].

LookOut [15] 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 [27]). The main algorithmic steps of LookOut are depicted via an example in Figure 5. 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 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 5, 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 $\{F2, F3\}$ 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) [17] 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 6-a) to -c) illustrate three subspaces with correlated features ($\{F0, F1\}$, $\{F0, F1, F8\}$ and $\{F11, F12, F13\}$) while Figure 6-d) a subspace with non correlated features ($\{F11, F12\}$).

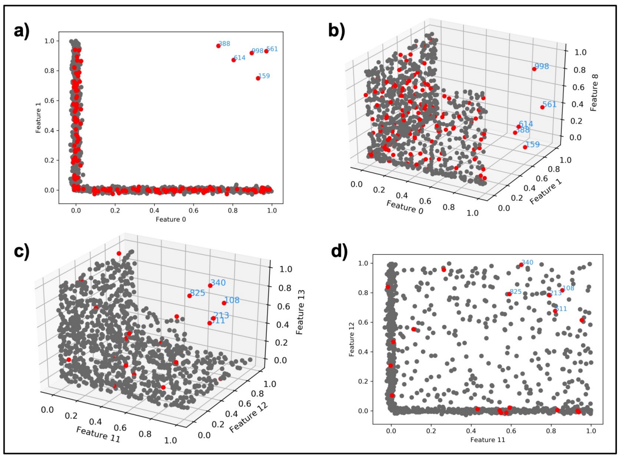


Figure 6: Data distribution in augmented/projected subspaces of HiCS Datasets

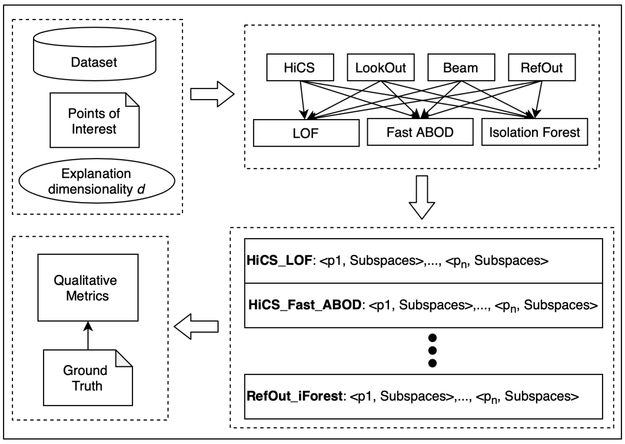
Subspace contrast in HiCS is measured using two-sample statistical tests[2] which are applied to the raw feature values under the null hypothesis that *both samples originate from the same underlying probability density function.* To enhance statistical precision, HiCS performs the statistical test for 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 correlated features seems effective for highly clustered anomalies, correlated subspaces may not always explain outliers, as depicted in Figure 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.

3 BENCHMARKING ENVIRONMENT

The algorithms along with the datasets used in our testbed are available in our GitHub repository[3] to ensure repeatability of our experiments. Regarding outlier detectors, we used the implementation of LOF and iForest from Scikit-learn [30] and Fast ABOD from PyOD [50]. We have implemented LookOut, RefOut and Beam in java and modified HiCS implementation from ELKI [37]. 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.



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 7: Pipelines of outlier detectors & explainers

3.1 Pipelines of Executed Algorithms

As illustrated in Figure 7, 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 3.3.

Regarding the choice of outlier detectors, we included in our testbed only LOF, Isolation Forest and Fast ABOD as representative of three widely used families of batch detection algorithms namely density, isolation and angle based outlier detection. As reported by several experimental studies [6, 8, 13] these algorithms frequently outperform distance or cluster-based algorithms in real and synthetic datasets while they do not require a thorough tuning of their hyper-parameters. Note also that experimentation with supervised detectors was outside the scope of our work due to the scarcity of labels regarding outlier/inlier data points.

To be able to retrieve the explaining subspaces for a number of outliers given as input, LookOut, Beam and RefOut heavily depend on the scores assigned by the detector in subspaces of different dimensionality explaining the given outliers. To shed some light regarding whether the explainers retrieve the relevant subspaces per outlier in practical settings, we employed unsupervised detectors that are not very sensitive to their hyper-parameter tuning. 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. These hyper-parameter values have been used in related experimental studies as [6], and allow us to detect the outliers in all datasets of our testbed. Hence, we can draw valuable conclusions for the subspace search techniques of explainers rather than the quality of the employed detector.

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

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

[3] https://git.io/JvuO6

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 1: Characteristics of real and synthetic datasets

3.2 Real and Synthetic Datasets

In this section we describe the real and synthetic datasets used in our testbed. The main challenge in explaining outliers stems from the exponential search space of feature subspaces rather than the size of the dataset. The difference in the execution time of explainers depends more on the pruning strategy they employ to enumerate subspaces rather than on the time spent by detectors to score the explored subspaces. The selected datasets are suitable for assessing the quality of outlier explanation algorithms, as they provide the gold standard regarding the subspace(s) explaining each anomaly. 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 4). The main characteristics of our datasets are summarized in Table 1.

Breast, Breast Diagnostic and *Electricity Meter* are real-world datasets widely used to benchmark ML methods for anomaly detection [9]. To facilitate comparison with already published results, we used the version of these datasets[4] made available by the authors of RefOut algorithm [18]. 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 [18] revealed that the reported outliers are *full space*. To obtain the best subspaces explaining them[5], we followed the method as described in [18] 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 2*d* 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 challenge summarization algorithms (HiCS and LookOut) as subspaces can best explain one outlier on average, e.g., for Electricity there are 1.43 outliers explained per relevant subspace (see Table 1).

HiCS synthetic datasets[6] were created by the authors of the HiCS [17] algorithm featuring *subspace outliers*. They initially splitted the datasets into 2*d* up to 5*d* 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 (100*d*) and splitted it into five subdatasets from 14 up to 100 dimensions. The ratio of relevant features is depicted in Table 1 ordered from low (14*d*) to high (100*d*) number of features. Note that every dataset contains 1000 points. As illustrated in Figure 8 and Table 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. An example of a 2*d* and a 3*d* relevant subspace is illustrated in Figures 6-a) and -c).

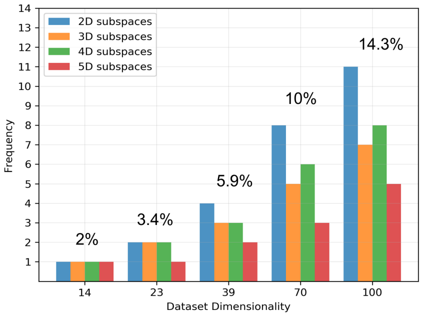


Figure 8: Dimensionality of subspaces relevant to outliers and contamination ratio of HiCS datasets

Note that the vast majority (∼ 91%) of outliers in HiCS datasets is explained by one subspace and few outliers (∼ 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 6-a) and -b) and (v) outliers are mixed with inliers in *lower dimensional projections* of relevant subspaces (see example of Figures 6-c) and -d). Note that all outliers in HiCS datasets can be discovered by the three detectors used in our testbed.

3.3 Evaluation Metric

In this section we present the metric used to evaluate the effectiveness of the 12 pairs of outlier detection and explanation

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

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

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

algorithms (see Figure 7). 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 $EXP_a(p)$. The first metric we used is Mean Recall (see Eq. **??**) of an explainer a for a set of points P, assessing how many relevant subspaces were returned on average by a for every point $p \in P$. Note that a subspace in $EXP_a(p)$ is considered relevant only if it is a member of REL_p. I.e., a subspace in $EXP_a(p)$ is considered relevant for a point p only if it is identical with a subspace in REL_p. We consider only the Recall (see Eq. **??**) of points that are explained at a given dimensionality according to the ground truth. As described in Section 3.2, every point has very few relevant subspaces in our datasets. Thus high Mean Recall means that the explainer was able to exploit the relevant subspaces for the majority of the points explained at a given dimensionality. 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 [40]. To compute MAP of an explainer a for a set of points P, we initially compute the precision (see Eq. 1) which is used to compute the Average Precision (see Eq. 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. 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. On the contrary, binary metrics like accuracy, precision and recall do not account for the ordering of the results. Note that [6, 8] use average precision to assess the quality of the outlier detector while [28] uses precision and recall to assess the explanation quality. To the best of our knowledge, it is the first work that relies on MAP to assess effectiveness of the subspace search strategies of different explainers.

$$\text{Precision}_a(p) = \frac{|REL_p \cap EXP_a(p)|}{|EXP_a(p)|} \tag{1}$$

$$\text{AveP}_a(p) = \frac{\sum_{k=1}^{|EXP_a(p)|} \text{P@k}(p) * rel(k)}{|REL_p|} \tag{2}$$

$$\text{MAP}_a(P) = \frac{1}{|P|} \sum_{p \in P} \text{AveP}(p) \tag{3}$$

4 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 [18] as well as the synthetic datasets used in the evaluation of HiCS [17]. 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 [28] running on HiCS [17] datasets. In our study, we incorporate three more explainers, namely RefOut [18], HiCS [17] and LookOut [15], using also real world datasets. Moreover, in contrast to [28] we formulate different trade-offs by evaluating the pruning strategies under different explanation sizes as well as full and sub-space outliers.

4.1 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 3.1 for the synthetic and real-world datasets presented in Section 3.2. Figure 9 depicts for each dataset, the MAP (y-axis) of different outlier detection and explanation pipelines for explanations of increasing dimensionality (x-axis).

Figures 9-a) to -e) illustrate the MAP obtained in the five synthetic datasets of our testbed. Starting from the 14 dimensions in Figure 9-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 not retrieve all the relevant subspaces. Passing to 23 dimensions in Figure 9-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 6). 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 9-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 (see Section 3.1) for every subspace considered

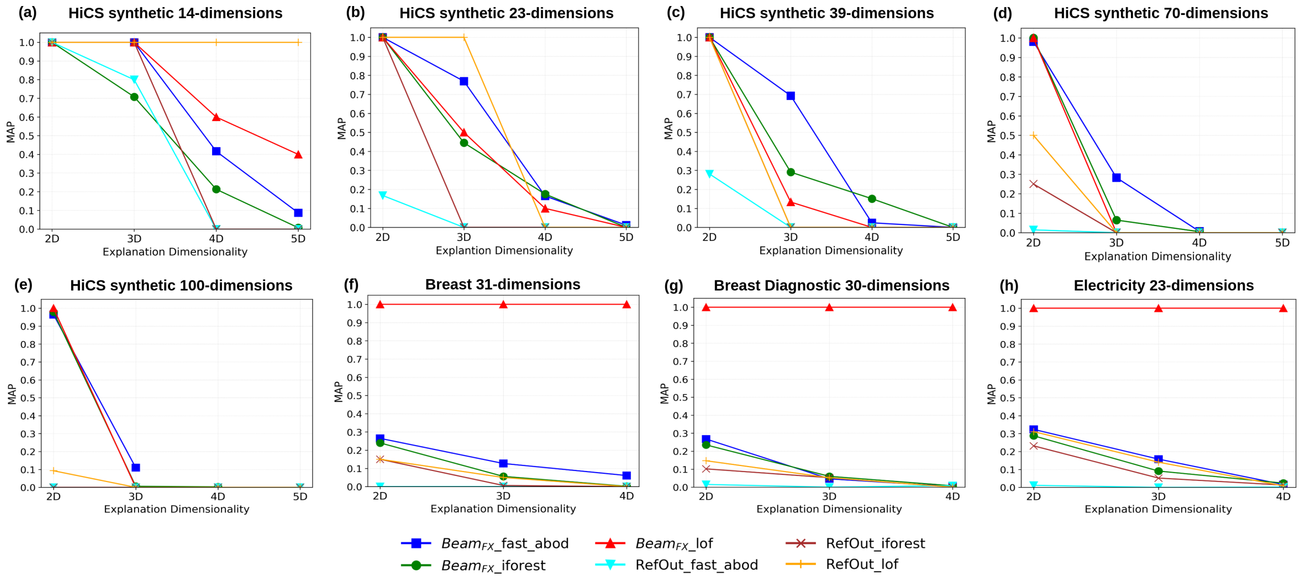


Figure 9: Mean Average Precision (MAP) of Beam and RefOut in HiCS synthetic datasets (a)-(e) and real-world datasets (f)-(h) for explanations of increasing dimensionality (best viewed in color)

by Beam up to 4*d* explanations for 70*d* and 100*d* datasets and Fast ABOD up to 4*d* explanations in 70*d* and up to 3*d* in 100*d* datasets. Specifically, to explain 100 outliers with 5*d* explanations in a 70*d* dataset, Beam needs to assess approximately 2.2M subspaces. In Section 4.3, we demonstrate that Beam requires an efficient detector such as LOF to assess a significant amount of subspaces.

Figures 9-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., 4*d* and 5*d*) and high dataset dimensionality (e.g., 70*d* and 100*d*). 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.

4.2 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 3.1 for the synthetic and real-world datasets presented in Section 3.2. Figure 10 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 10-a) to -e) show the MAP of different algorithms for the five synthetic datasets of our testbed. Starting from 14 dimensions in Figure 10-a), HiCS and LookOut with LOF achieve optimal MAP regardless of the explanation dimensionality. As dataset's dimensionality and outlier ratio increase in Figures 10-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 4*d* 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 10-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

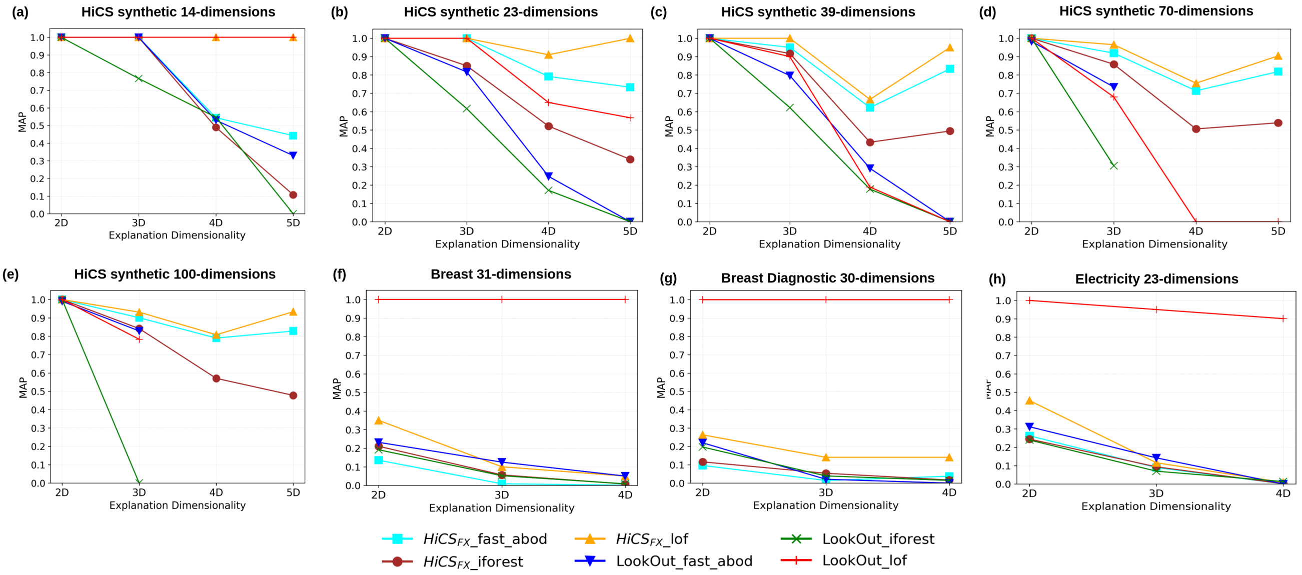


Figure 10: Mean Average Precision (MAP) of HiCS and LookOut in HiCS synthetic datasets (a)-(e) and real-world datasets (f)-(h) for explanations of increasing dimensionality (best viewed in color)

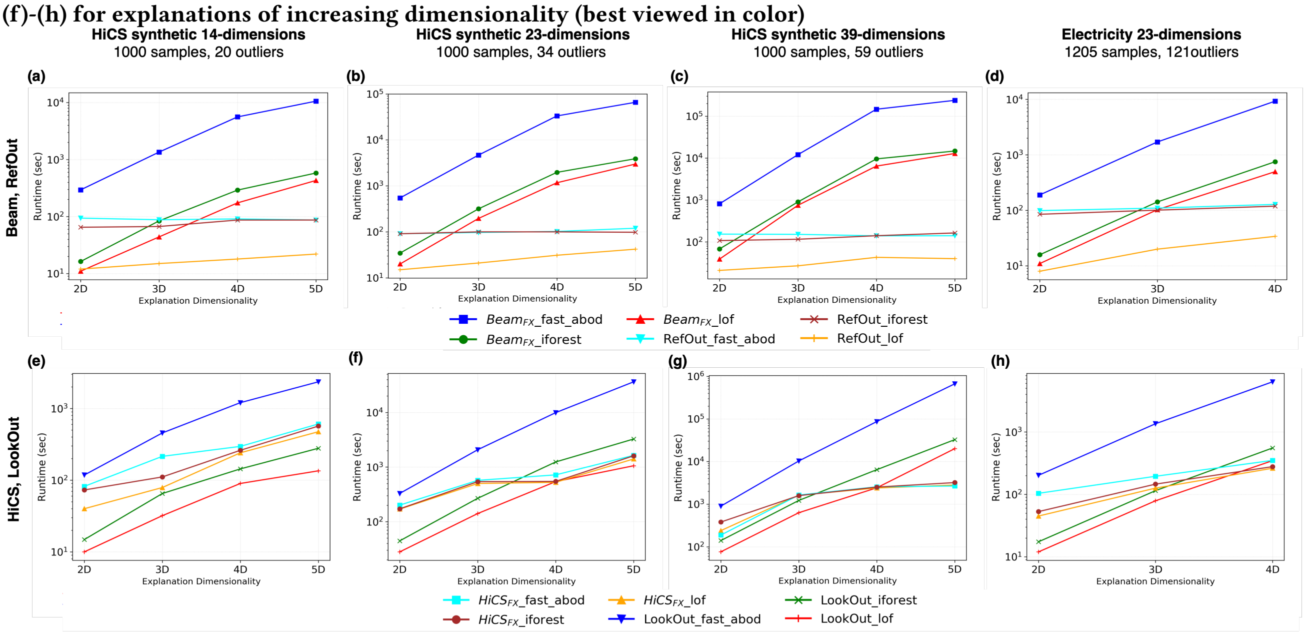


Figure 11: Runtime of detection and explanation pipelines (best viewed in color)

subspaces. According to complementary experiments not presented here due to space restrictions, 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 4.3, we demonstrate that LOF is the most efficient detector when a significant amount of subspaces need to be assessed.

Figures 10-f) to -h) illustrate the MAP obtained in the 3 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 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. 14*d*). 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.

4.3 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 4.1 and 4.2. In this respect, we are using the same synthetic (up to HiCS 39*d*) and real (Electricity 23*d*) datasets containing a similar amount of samples (∼ 1000). We report execution time only for Electricity 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 (2-5*d*) 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 11, 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 [8] for the same hyper-parameter settings as those used in our testbed (see Section 3.1). 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 11-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 39*d* datasets and 2*d* 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 39*d* datasets and 5*d* explanations.

Explanation Summarization. The runtime of pipelines involving LookOut and HiCS are illustrated in Figures 11-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[7] outperforms all HiCS pipelines up to 4*d* explanations (by 1 order of magnitude in 2*d*). 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 39*d* dataset for 5*d* explanations.

Table 2 demonstrates the point explanation and summarization algorithms along with their corresponding detector that exhibit the best tradeoff between effectiveness (according to Figures 9 and 10) and efficiency (according to Figure 11) from 2*d* up to 5*d* 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 10-c), while they have the same execution time in Figure 11-g). In cells 2*d* and 3*d* 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 5*d* 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 5*d* explanations. The main conclusions drawn from Table 2 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 (3*d* - 4*d*) 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 3*d* (for 21% ratio).

3. *Exhaustive subspace search* employed by LookOut exhibits top effectiveness and efficiency for full space outliers regardless of the explanation dimensionality, as well as, for subspace outliers up to 3*d*. Its effectiveness significantly drops for subspace outliers explained by subspaces greater than 3*d* (for 21% ratio).

4. *Correlation heuristic* exploited by HiCS achieves the best tradeoff for 4*d*-5*d* 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.

5 RELATED WORK

In this section we survey additional explanation algorithms for data in rest (databases) or in motion (streams) and justify why they have not included in our benchmark.

Explaining Black-Box Models. Several methods have been proposed to explain why a supervised model predicted a particular label for a particular example [10, 19, 24, 26, 33]. LIME [33] constructs a linear interpretable model that is locally faithful to the

[7] LookOut has been experimentally evaluated by its authors [15] only with iForest and 2*d* explanations.

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 2: Tradeoffs of outlier detection and explanation algorithms

predictor. [10, 19] explain the model by perturbing the features to quantify their influence on predictions. Other works aim to produce explanations in the form of feature relevance scores by comparing the difference between a classifier's prediction score and the score when a feature is assumed to be unobserved [34], or by considering the local gradient of the classifier's prediction score with respect to the features for a particular example [4]. [38, 39] considered how to score features in a way that takes into account the joint influence of feature subsets on the classification score. This body of work requires 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 [12].

Explaining Outliers in Query Answers. Scorpion [47] 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 [7]. 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 [35]. It relies on a set of explanation templates given by analysts that need to be precomputed per 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.

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 [16] 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 [45]. Finally, EXstream [49] 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., =, <, ≤) 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 [15]. 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 [45].

6 CONCLUSIONS AND FUTURE WORK

In this experimental study, we addressed missing insights regarding the performance of existing outlier 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 [17]) or (ii) of scores given by an outlier detector in subspaces (LookOut [15], Beam [28] and RefOut [18]). The former strategy is effective when highly clustered outliers over correlated features are contained in datasets regardless of their dimensionality, while the latter is effective in low explanation dimensionality where the outlier detectors can discriminate accurately the outliers 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 outliers in LOF, ABOD and iForest, is actually decoupled from the search of subspaces likely to contain them. HiCS, RefOut and Beam instead are *explaining outlier detectors* that rely on per-subspace measures to quantify the explanation quality of subspaces. We are planning to extend our testbed with recent works [44] 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 [25].

Another interesting aspect would be to investigate outlier explanation in stream processing settings such as LODA [31].

We should finally stress that existing outlier explanation and summarization algorithms actually provide *descriptive explanations*. In essence, subspace explanations are verbose descriptions of the decision boundary discovered by unsupervised detectors to distinguish inliers from outliers. This is the reason why explanation tasks should be re-executed for every new bunch of data made available in ML pipelines even if they stem from the same generative process. As summarization algorithms can only exploit the subspaces which are assessed to be relevant to a given set of points, they may result in summaries of very poor quality when individual outliers are explained by disjoint feature subsets. In this respect, we are planning to build a surrogate model to predict the scores (or labels) of points produced by an unsupervised outlier detector and approximate its decision boundary using minimal predictive signatures. Such *predictive explanations* overcome the high computation cost of subspace search per point and provide formal guarantees regarding minimality in the explanation dimensionality.

REFERENCES

[1] F. Abuzaid, P. Bailis, J. Ding, E. Gan, S. Madden, D. Narayanan, K. Rong, and S. Sahaana. Macrobase: Prioritizing attention in fast data. *ACM Trans. Database Syst.*, 43(4):15:1–15:45, Dec. 2018.

[2] C. C. Aggarwal. *An Introduction to Outlier Analysis*, pages 1–40. Springer New York, 2013.

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

[4] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K. Müller. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831, 2010.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *SIGMOD Conference*, 2000.

[6] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenkova, E. Schubert, I. Assent, and M. 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.

[7] T. Dasu, J. M. Loh, and D. 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.

[8] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, 74:406 – 421, 2018.

[9] D. Dua and C. Graff. UCI machine learning repository, 2017.

[10] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, pages 3449–3457, 2017.

[11] J. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2001.

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

[13] X. Gu, L. Akoglu, and A. Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. In *NeurIPS*, pages 10921–10931, 2019.

[14] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), Aug. 2018.

[15] N. Gupta, D. Eswaran, N. Shah, L. Akoglu, and C. Faloutsos. Beyond outlier detection: Lookout for pictorial explanation. In *ECML/PKDD*, 2018.

[16] V. Jeyakumar, O. Madani, A. Parandeh, A. Kulshreshtha, W. Zeng, and N. 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.

[17] F. Keller, E. Müller, and K. Böhm. Hics: High contrast subspaces for density-based outlier ranking. *ICDE*, pages 1037–1048, 2012.

[18] F. Keller, E. Müller, A. Wixler, and K. Böhm. Flexible and adaptive subspace search for outlier analysis. In *CIKM*, 2013.

[19] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *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.

[20] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *PAKDD*, 2009.

[21] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *KDD*, pages 444–452. ACM, 2008.

[22] 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.

[23] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[24] S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In *NeurIPS*, pages 4765–4774, 2017.

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

[26] G. Montavon, W. Samek, and K. Müller. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.*, 73:1–15, 2018.

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

[28] X. V. Nguyen, J. Chan, S. Romano, J. Bailey, C. Leckie, K. Ramamohanarao, and J. Pei. Discovering outlying aspects in large datasets. *Data Mining and Knowledge Discovery*, 30:1520–1555, 2016.

[29] A. Nurunnabi and G. 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.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

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

[32] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich. Data lifecycle challenges in production machine learning: A survey. *SIGMOD Rec.*, 47(2):17–28, 2018.

[33] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *KDD*, pages 1135–1144, 2016.

[34] M. Robnik-Sikonja and I. Kononenko. Explaining classifications for individual instances. *TKDE*, 20:589–600, 2008.

[35] S. Roy, L. Orr, and D. Suciu. Explaining query answers with explanation-ready databases. *Proc. VLDB Endow.*, 9(4):348–359, Dec. 2015.

[36] S. Sathe and C. C. Aggarwal. Subspace outlier detection in linear time with randomized hashing. *ICDM*, pages 459–468, 2016.

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

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

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

[40] W. Su, Y. Yuan, and M. Zhu. A relationship between the average precision and the area under the roc curve. In *ICTIR '15*, 2015.

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

[42] K. M. Ting, T. Washio, J. R. Wells, and S. Aryal. Defying the gravity of learning curve: a characteristic of nearest neighbour anomaly detectors. *Machine Learning*, 106:55–91, 2016.

[43] L. Tran, L. Fan, and C. Shahabi. Distance-based outlier detection in data streams. *Proc. VLDB Endow.*, 9(12):1089–1100, Aug. 2016.

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

[45] I. Tsamardinos, G. Borboudakis, P. Katsogridakis, P. Pratikakis, and V. Christophides. A greedy feature selection algorithm for big data of high dimensionality. *Machine Learning*, 108(2):149–202, 2019.

[46] B. L. Welch. The significance of the difference between two means when the population variances are unequal. *Biometrika*, 29(3/4):350–362, 1938.

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

[48] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar. Enhancing data analysis with noise removal. *TKDE*, 18(3):304–319, Mar. 2006.

[49] H. Zhang, Y. Diao, and A. Meliou. Exstream: Explaining anomalies in event stream monitoring. In *EDBT*, pages 156–167, Mar. 2017.

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

[51] A. Zimek and P. Filzmoser. There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(6), 2018.