Explaining Anomalies in Groups with Characterizing Subspace Rules

Meghanath Macha · Leman Akoglu

Abstract Anomaly detection has numerous applications and has been studied vastly. We consider a complementary problem that has a much sparser literature: anomaly *description*. Interpretation of anomalies is crucial for practitioners for sense-making, troubleshooting, and planning actions. To this end, we present a new approach called x-PACS (for eXplaining Patterns of Anomalies with Characterizing Subspaces), which "reverse-engineers" the known anomalies by identifying (1) the groups (or patterns) that they form, and (2) the characterizing subspace and feature rules that separate each anomalous pattern from normal instances. Explaining anomalies in groups not only saves analyst time and gives insight into various types of anomalies, but also draws attention to potentially critical, repeating anomalies. In developing x-PACS, we first construct a desiderata for the anomaly description problem. From a descriptive data mining perspective, our method exhibits five desired properties in our desiderata. Namely, it can unearth anomalous patterns (*i*) of multiple different types, (*ii*) hidden in arbitrary subspaces of a high dimensional space, (*iii*) interpretable by human analysts, (*iv*) different from normal patterns of the data, and finally (*v*) succinct, providing a short data description. No existing work on anomaly description satisfies all of these properties simultaneously. Furthermore, x-PACS is highly parallelizable; it is linear on the number of data points and exponential on the (typically small) largest characterizing subspace size. The anomalous patterns that x-PACS finds constitute interpretable "signatures", and while it is not our primary goal, they can be used for anomaly detection. Through extensive experiments on real-world datasets, we show the effectiveness and superiority of x-PACS in anomaly explanation over various baselines, and demonstrate its competitive detection performance as compared to the state-of-the-art.

Meghanath Macha and Leman Akoglu
Heinz College, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
E-mail: {meghanam, lakoglu}@andrew.cmu.edu

1 Introduction

Given a large dataset containing normal and labeled anomalous points, how can we *characterize* the anomalies? What combinations of features and feature values make the anomalies stand out? Are there *anomalous patterns*, that is, do anomalies form *groups*? How many different types of anomalies (or groups) are there, and how can we *describe* them succinctly for downstream investigation and decision-making by analysts?

Anomaly mining is important for numerous applications in security, medicine, finance, etc., for which many *detection* methods exist [1]. In this work, we consider a complementary problem to this vast body of work: the problem of anomaly *description*. Simply put, we aim to find human-interpretable explanations to already identified anomalies. Our goal is "reverse-engineering" known anomalies by unearthing their hidden characteristics—those that make them stand out.

The problem arises in a variety of scenarios, in which we obtain labeled anomalies, albeit no description of the anomalies that could facilitate their interpretation. Example scenarios are those where

(1) the detection algorithm is a "black-box" and only provides labels, due to intellectual property or security reasons (e.g., Yelp's review filter [43]),

(2) the detection algorithm does not produce an interpretable output and/or cannot explicitly identify anomalous patterns (e.g., ensemble detectors like bagged LOF [36] or isolation forest [38]), and

(3) the anomalies are identified via external mechanisms (e.g., when software or compute-jobs on a cluster crash, loan customers default, credit card transactions get reported by card owners as fraudulent, products get reported by consumers as faulty, etc.). This setting also arises when security experts set up "honeypots" to attract malicious users, and later study their operating mechanisms (often manually). Examples include fake followers of honeypot Twitter accounts [37] and fraudulent bot-accounts that click honeypot ads [12].

Explaining anomalies is extremely useful in practice as anomalies are to be investigated by human analysts in almost all scenarios. Interpretation of the anomalies help the analysts in sense-making and knowledge discovery, troubleshooting and decision making (e.g., planning and prioritizing actions), and building better prevention mechanisms (e.g., policy changes).

Our work taps into the gap between anomaly detection and its end usage by analysts, and introduces x-PACS for *characterizing the anomalies in high-dimensional datasets*. Our emphasis is explaining the anomalies *in groups*[1]. We model the anomalies to consist of (*i*) various patterns (i.e., sets of clustered anomalies) and (*ii*) outliers (i.e., scattered anomalies different from the rest). For example in fraud, malicious agents that follow similar strategies, or those who work together in "coalition", exhibit similar properties and form anomalous groups. Bots deployed for e.g., click or email spam also tend to produce similar footprints as they follow the same source of command-and-control. At the same time, there may be multiple groups of fraudsters or bots with different strategies.

Explaining anomalies in groups has three key advantages: (1) it saves investigation time by providing a compact explanation, rather than the analyst having to

---

[1] In this text, phrases 'anomalous pattern', 'clustered anomalies', and 'group of anomalies' are interchangeable.

go through anomalies one by one, (2) it provides insights into the characteristics of different anomaly types, and (3) importantly, it draws attention to anomalies that form patterns, which are potentially more critical as they are repetitive.

To lay out the challenges from a data mining perspective, we first introduce a list of desired properties (Desiderata 1–5) that approaches to the problem of anomaly explanation should satisfy. We then summarize our contributions.

1.1 Desiderata for Anomaly Description

In a nutshell, anomaly explanation methods should effectively characterize different kinds of anomalies present in the data, handle high dimensional datasets, and produce human-interpretable explanations that are distinct from normal patterns as well as succinct in length.

D1 **Identifying *different types* of anomalies:** Anomalies are generated by mechanisms other than the normal. Since such mechanisms can vary (e.g., different fraud schemes), it is likely for the anomalies to form multiple patterns in potentially different feature subspaces. A description algorithm should be able to characterize all types of anomalies.

D2 **Handling *high-dimensionality*:** Data instances typically have tens or even hundreds of features. It is meaningful to assume that the anomalies in a pattern exhibit only a (small) fraction of features in common. In other words, anomalies are likely to "hide" in sparse subspaces of the full space.

D3 ***Interpretable* descriptions:** It is critical that the explanation of the anomalies can be easily understood by analysts. In other words, descriptions should convey what makes a group of instances anomalous in a human-interpretable way.

D4 ***Discriminative (or detection)* power:** Explanations of anomalies should not also be valid for normal points. In other words, descriptions should be discriminative and separate the anomalies from the normal points sufficiently well. As a result, they could also help detect future anomalies of the same type.

D5 ***Succinct* descriptions:** It is particularly important to have simple and concise representations, for ease of visualization and avoiding information overload. This follows the Occam's razor principle.

1.2 Limitations of Existing Techniques

Providing interpretable explanations for anomalies is a relatively new area of study compared to anomaly detection. However, the problem has similarities to description based techniques for imbalanced datasets. Almost all existing work in the anomaly detection literature assume anomalies to be scattered, and try to explain them one at a time [10, 11, 24, 27, 34, 50]. Related work on collective data description [18, 55], including rare class characterization [20, 21], assume a single pattern and/or do not look for subspaces. Other closely related areas are subgroup discovery and inductive rule learners. Subgroup discovery techniques [16, 22, 25, 26, 39, 56] aim to describe individual classes while inductive rule learners [9, 8, 13, 15, 19] focus on describing multiple classes with the aim of generalization rather than explanation. Another related line of work involves techniques for explaining black-box classifiers [14, 28, 42, 51] where the emphasis is on explaining the prediction of

a single instance rather than a group of instances. Moreover, none of these work has an explicit emphasis on succinct, minimal descriptions.

To the best of our knowledge and as we expand in related work in §5, there is no existing work that provides a principled and general approach to the anomaly description problem that meet all of the goals in our desiderata adequately. (See Table 9 for an overview and comparison of related work.)

1.3 Summary of Contributions

Our work sets out to fill the gap, with the following main contributions:

- **Desiderata for Anomaly Description:** We introduce a new desiderata and target five rules-of-thumb (D1–D5) for designing our approach.
- **Description-in-Groups (DIG) Problem:** We formulate the explanation problem as one of identifying the various groups that the anomalies form (D1) within low-dimensional subspaces (D2).
- **Description Algorithm** X-PACS**:** We introduce a new algorithm that produces interpretable rules (D3)—intervals on the features in each subspace, that are also discriminative (D4)—characterizing the anomalies in the group within a subspace but as few normal points as possible.
- **A New Encoding Scheme:** We design a new encoding-based objective for describing the anomalies in groups, based on the minimum description length (MDL) principle [52]. Through non-monotone submodular optimization we carefully select the minimal subspace rules (D5) that require the fewest 'bits' to collectively describe all the anomalies.

Reproducibility: All of our code and datasets are open-sourced at https://github.com/meghanathmacha/xPACS.

2 Overview and Problem Statement

In a nutshell, *our goal is to identify a few, small micro-clusters of anomalies hidden in arbitrary feature subspaces that collectively and yet succinctly represent the anomalies and separate them from the normal points.* Specifically, our proposed X-PACS finds a small set of low-dimensional hyper-ellipsoids (i.e., micro-clusters corresponding to *anomalous patterns* each enclosing a subset of the anomalies), and reveals scattered anomalies (i.e., outliers not contained in any ellipsoid).

Features that are part of the subspace in which a hyper-ellipsoid lies constitute its *characterizing subspace*. Ranges of values these features take are further characterized by the location (center and radii) of a hyper-ellipsoid within the subspace. Each hyper-ellipsoid is simply a "*pack* of anomalies" (hence the name X-PACS[2]). The rest of the paper uses 'hyper-ellipsoid' or '*pack*' in reference to anomalous patterns.

---

[2] X- refers to the number of packs, which we automatically identify via our data encoding scheme (§3.3). We use this naming convention after X-MEANS [49], which finds the number of k-means clusters automatically in an information-theoretic way.
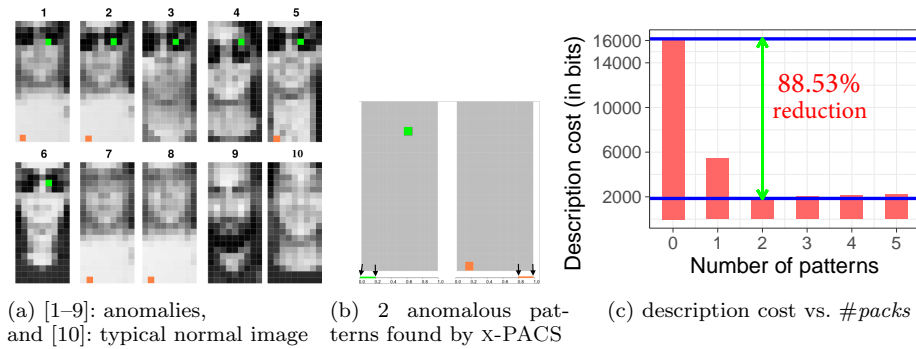
(a) [1–9]: anomalies, and [10]: typical normal image

(b) 2 anomalous patterns found by x-PACS

(c) description cost vs. #packs

Fig. 1: (best in color) Example x-PACS input–output.

2.1 Example x-PACS input-output

In Fig. 1 we show an example of the input and output of x-PACS. We consider the face images dataset, in which x-PACS identifies a minimum-description packing with two anomalous patterns and an outlier. In Fig. 1a, we visualize the dataset, where pixels are dimensions/features, that contains 9 labeled 'anomalies': [images 1–8] of 2 types (people w/ sunglasses or people w/ white t-shirt or both) + [image 9] an outlier (one person w/ beard). We also show [image 10], which is representative of 82 normal samples (people w/ black t-shirt w/out beard or sunglasses). In Fig. 1b, we display the anomalous patterns found by x-PACS (characterizing subspaces are 1-d, feature rules/intervals shown at the bottom with arrows—the smaller, the darker the pixel) together explain anomalies 1–8 succinctly; 1-d pack (left) encloses images {1–6}, 1-d pack (right) encloses images {1,2,5,7,8}. Corresponding features/pixels highlighted on enclosed images. In Fig. 1c, we plot the description length (in bits, see §3.3.1) of anomalies individually (0 packs), vs. w/ 1–5 packs. x-PACS automatically finds the best number of patterns (=2) that describe the anomalies and reveals the (unpacked) outlier [image 9].

2.2 Main Steps

Our x-PACS consists of three main steps, each aiming to meet various criteria in our desiderata (D1–D5).

1. First we employ subspace clustering to automatically identify multiple clusters of anomalies (D1) embedded in various feature subspaces. Advantages of subspaces are two-fold: handling "curse of dimensionality" (D2) and explaining each pattern with only a few features (D5).

2. In the second step, we represent anomalies in each subspace cluster by an axis-aligned hyper-ellipsoid. Ellipsoids, in contrast to hyperballs, allow for varying spread of anomalies in each dimension. Axis-alignment ensures interpretable explanation with original features, which typically have real meaning to a user (D3). Moreover, we introduce a convex formulation to ensure that the ellipsoids are "pure" and enclose very few non-anomalous points, if at all, such that the characterization is discriminative (D4).

3. Final step is summarization, where we strive to generate minimal descriptions for ease of comprehension (D5). To decide which patterns describe the anomalies most succinctly, we introduce an encoding scheme based on the Minimum Description Length (MDL) principle [52]. Our encoding-based objective lends itself to non-monotone submodular function maximization. Using an algorithm with approximation guarantees we identify a short list of patterns (hyper-ellipsoids) that are (*i*) compact with small radii, i.e., range of values that anomalies take per feature in the characterizing subspace is narrow; (*ii*) non-redundant, which "pack" (i.e., enclose) mostly different anomalies in various subspaces, and (*iii*) pure, which enclose either none or only a few normal points. Importantly, the necessary number of packs is automatically identified based on the MDL criterion.

Remark: Note that while x-PACS identifies descriptive patterns of the anomalies, those can also be used for detection. Each pattern, along with its characterizing features and its enclosing boundary within that subspace can be seen as a discriminative *signature* (or set of rules), and can be used to label future instances—a new instance that falls within any of the *packs* is labeled as anomalous. Instead of a single signature or an abstract classifier function or model, however, x-PACS identifies multiple, interpretable signatures.

2.3 Notation and Definitions

Input dataset is denoted with $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, containing m points in d-dimensions, where $\mathcal{F}$ depicts the feature set. A subset $\mathcal{A} \subset \mathcal{D}$ of points are labeled as $y_{\mathcal{A}} = $ 'anomalous', $|\mathcal{A}| = a$. The rest are $y_{\mathcal{D} \setminus \mathcal{A}} = $ 'normal' points, denoted by $\mathcal{N}$, $|\mathcal{N}| = n$, $a + n = m$.

Our goal is to find "enclosing shapes", called *packs*, that collectively contain as many of the anomalies as possible. While arbitrary shapes would allow for higher flexibility, we restrict these shapes to the hyper-ellipsoids family for ease of interpretation. This is not a strong limitation, however, since anomalous patterns are expected to form compact micro-clusters in some feature subspaces, rather than lie on arbitrarily shaped manifolds. A *pack* is formally defined as follows.

Definition 1 (pack) A *pack* p_k is a hyper-ellipsoid in a feature subspace $\mathcal{F}_k \subseteq \mathcal{F}$, $|\mathcal{F}_k| = d_k$, characterized by its center $\mathbf{c}_k \in \mathbb{R}^{d_k}$ and matrix $\mathbf{M}_k \in \mathbb{R}^{d_k \times d_k}$ where

$$p_k(\mathbf{c}_k, \mathbf{M}_k) = \{\mathbf{x} | (\mathbf{x} - \mathbf{c}_k)^T \mathbf{M}_k^{-1} (\mathbf{x} - \mathbf{c}_k) \leq 1\} .$$

We denote the anomalies that p_k encloses by $\mathcal{A}_k \subseteq \mathcal{A}$, and the normal points that it encloses by $\mathcal{N}_k \subset \mathcal{N}$.

Definition 2 (packing) A *packing* $\mathcal{P}$ is a collection of *packs* as defined above; $\mathcal{P} = \{p_1(\mathbf{c}_1, \mathbf{M}_1), \ldots, p_K(\mathbf{c}_K, \mathbf{M}_K)\}$ with size K.

2.4 Problem Statement

Based on the above definitions, our description-in-groups problem is formally:

Problem 1 Given a dataset $\mathcal{D} \in \mathbb{R}^{m \times d}$ containing a anomalous points in $\mathcal{A}$ and n non-anomalous or normal points in $\mathcal{N}$, $a \ll n$;

Find a set of anomalous patterns (*packs*) $\mathcal{P} = \{p_1, p_2, \ldots, p_K\}$, each containing/enclosing a subset of the anomalies $\mathcal{A}_k$, where $\bigcup_{1 < k \leq K} \mathcal{A}_k \subseteq \mathcal{A}$,

such that $\mathcal{P}$ provides the minimum description length $L(\mathcal{A}|\mathcal{D}, \mathcal{P})$ (in bits) for the anomalies in $\mathcal{D}$. (We introduce our MDL-based encoding scheme and cost function $L(\cdot)$ later in §3.3.)

Note that while packs enclose different subsets of anomalies in general, any two packs can have some anomalous points in common (since an anomaly can be explained in different ways), i.e., $\mathcal{A}_k \cap \mathcal{A}_l \neq \emptyset \quad \exists k, l$. Packs can also share common features in their subspaces (as different types of anomalies may share some common characteristics), i.e., $\mathcal{F}_k \cap \mathcal{F}_l \neq \emptyset \quad \exists k, l$. Moreover, the enclosing boundary of a *pack* may also contain some non-anomalous points. These issues related to the redundancy and purity of the packs would play a key role in the "description cost" of the anomalies. When it comes to identifying a small set of *packs* out of a list of candidates, we formulate an encoding scheme as a guiding principle to selecting the smallest, least redundant, and the purest collection of *packs* that would yield the shortest description of all the anomalies.

3 x-PACS: Explaining Anomalies in Groups

Next we present the details of x-PACS, which consists of three building blocks:

§3.1 **Subspace Clustering**: Identify clusters of anomalies in various subspaces

§3.2 **Refinement**: Transform box-like subspace clusters to pure and compact hyper-ellipsoids (or *packs*)

§3.3 **Summarization**: Select subset of *packs* that yields the minimum description length of anomalies

We present our algorithms for each of these next.

3.1 Subspace Clustering: Finding Hyper-rectangles

In our formulation, we allow for anomalies to form multiple patterns, intuitively each containing anomalies of a different kind. We model anomalous patterns as compact "micro-clusters" in various feature subspaces.

In the first step, we use a subspace clustering algorithm, similar to CLIQUE [3] and ENCLUS [7], that discovers subspaces with high-density anomaly clusters in a bottom-up, Apriori fashion. There are two main differences that we introduce. First, while prior techniques focus on a density (minimum count or mass) criterion, we use two criteria: (*i*) mass and (*ii*) purity, in order to find clusters that respectively contain many anomalous points, but also a low number of normal points. Second, we do not enforce a strict grid over the features but find varying-length high-density intervals through density estimation in a data-driven way.

Simply put, the search algorithm starts with identifying 1-dimensional intervals in each feature that meet a certain mass threshold. These intervals are then combined to generate 2-dimensional candidate rectangles. In general, k-dimensional

---

Algorithm 1 SUBCLUS $(\mathcal{D}, ms, \mu)$

Input: dataset $\mathcal{D} = \mathcal{A} \cup \mathcal{N} \in \mathbb{R}^{m \times d}$ with labeled anomalous and normal points, mass threshold
    $ms \in \mathbb{Z}$, purity threshold $\mu \in \mathbb{Z}$
Output: set of hyper-rectangles $\mathcal{R} = \{R_1, R_2, \ldots\}$ each containing min. ms anomalous &
    max. μ normal points
1: Let $\mathcal{R}^{(k)}$ denote k-dimensional hyper-rectangles. Initialize $\mathcal{R}^{(1)}$ by kernel density estima-
    tion with varying quantile thresholds in $q = \{80, 85, 90, 95\}$, set $k = 1$
2: **for each** hyper-rectangle $R \in \mathcal{R}^{(k)}$ **do**
3:     **if** mass$(R) \geq ms$ **then**
4:         **if** impurity$(R) \leq \mu$ **then** $\mathcal{R}^{(k)}_{pure} = \mathcal{R}^{(k)}_{pure} \cup R$
        else $\mathcal{R}^{(k)}_{\neg pure} = \mathcal{R}^{(k)}_{\neg pure} \cup R$
5:     **end if**
6: **end for**
7: $\mathcal{R} = \mathcal{R} \cup \mathcal{R}^{(k)}_{pure}$
8: $\mathcal{R}^{(k+1)} := \text{generateCandidates}(\mathcal{R}^{(k)}_{pure} \cup \mathcal{R}^{(k)}_{\neg pure})$
9: **if** $\mathcal{R}^{(k+1)} = \emptyset$ **then return** $\mathcal{R}$
10: $k = k + 1$, go to step 2

---

hyper-rectangles are generated by merging $(k-1)$-dimensional ones that meet the mass criterion in a hierarchical fashion. Thanks to the monotonicity property of mass, the search space is pruned efficiently. Hyper-rectangles generated during the course of the bottom-up algorithm that meet *both* the mass and purity criteria are reported as clusters. A hyper-rectangle is formally defined as follows.

Definition 3 (hyper-rectangle) Let $\mathcal{F} = f_1 \times f_2 \times \ldots \times f_d$ be our original d-dimensional numerical feature space. A hyper-rectangle $R = (s_1, s_2, \ldots, s_{d'})$, $d' \leq d$, resides in a space $f_{t_1} \times f_{t_2} \times \ldots \times f_{t_{d'}}$ where $t_i < t_j$ if $i < j$, and has d' sides, $s_z = [lb_z, ub_z]$, that correspond to individual intervals with lower and upper bounds in each dimension. A point $\mathbf{x} = \langle x_1, x_2, \ldots, x_d \rangle$ is said to be contained or enclosed in hyper-rectangle $R = (s_1, s_2, \ldots, s_{d'})$, if $lb_z \leq \mathbf{x}_{t_z} \leq ub_z$ $\forall z = \{1, \ldots, d'\}$.

    The outline of our subspace clustering is in Algorithm 1. It takes dataset $\mathcal{D}$ as input with anomalous and normal points, a mass threshold ms equal to the minimum number of required anomalous points and a purity threshold μ equal to the maximum number of allowed normal points to be contained inside, and returns hyper-rectangles that meet the desired criteria.

    To begin (line 1), we find 1-dimensional candidate hyper-rectangles, equivalent to intervals in individual features. To create promising candidate intervals initially, we find dense intervals with many anomalous points. To this end, we perform kernel density estimation (KDE[3]) on the anomalous points and extract the intervals of significant peaks.[4] This is achieved by extracting the contiguous intervals in each dimension with density larger than the q-th quantile of all estimated densities. q is varied in $[80, 95]$ to obtain candidate intervals of varying length. An illustration is given in Fig. 2. Since multiple peaks may exist, multiple intervals can be generated per dimension as q is varied.

---

[3] KDE involves two parameters - the number of points sampled to construct the smooth curve and the kernel bandwidth. We set the sample size to 512 points and use the Silverman's rule of thumb [54] to set the bandwidth.

[4] For categorical features, we would instead use histogram density estimation.
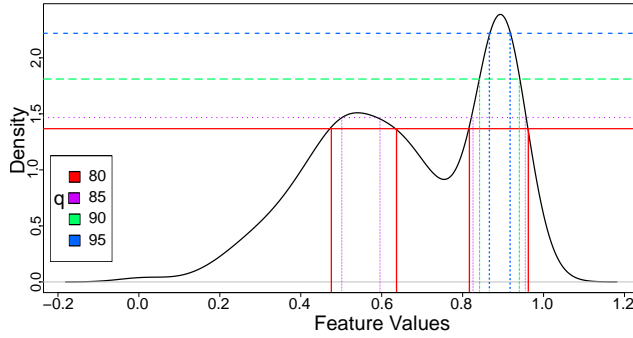
Fig. 2: Identifying candidate hyper-rectangles in 1-d (equivalent to intervals) by KDE for varying quantile thresholds q.

At any given level (or iteration) of the Apriori-like SUBCLUS algorithm, we scan all the candidates at that level (line 2–6) and filter out the ones that meet the mass criterion (line 3). Those that pass the filter are later merged to form candidates for the next level. Others with mass less than required are discarded, with no implications on accuracy. The correctness of the pruning procedure follows from the *downward closure property* of the mass criterion: for any k-dimensional hyper-rectangle with mass $\geq ms$, its projections in any one of $(k-1)$-dimensions must also have mass $\geq ms$.

At each level, we also keep track of the hyper-rectangles that meet both the mass and the purity criteria (line 4). Purity exhibits the *upward closure property*: for any $(k-1)$-dimensional hyper-rectangle that is pure (i.e. contains $\leq \mu$ normal points), any k-dimensional hyper-rectangle that subsumes it is also pure. This property could help us stop growing pure candidates by excluding them from the candidate generation step and speeding up the termination. While correct, however, such early-termination would prevent us from finding even purer hyper-rectangles later up in the hierarchy. To obtain as many candidate *packs* as possible, we continue our search for all hyper-rectangles that meet the mass criterion, and use the purity criterion for selecting the ones to be output (line 7).

The algorithm proceeds level by level. Having identified k-dimensional hyper-rectangles that satisfy the mass criterion, denoted $\mathcal{R}_{\geq ms}^{(k)} = \mathcal{R}_{pure}^{(k)} \cup \mathcal{R}_{\neg pure}^{(k)}$ (respectively for pure and not-pure sets), $(k+1)$-dimensional candidates are generated (line 8) in two steps: join and prune. The join step combines hyper-rectangles having first $(k-1)$ dimensions as well as sides in common. That is, if $(s_{u_1}, s_{u_2}, \ldots, s_{u_k})$ and $(s_{v_1}, s_{v_2}, \ldots, s_{v_k})$ are two k-dimensional hyper-rectangles in $\mathcal{R}_{\geq ms}^{(k)}$, we require $u_i = v_i$ and $s_{u_i} = s_{v_i} \ \forall i \in \{1, \ldots, (k-1)\}$ and $u_k < v_k$ to form candidate $(k+1)$-dimensional hyper-rectangles of the form $(s_{u_1}, s_{u_2}, \ldots, s_{u_k}, s_{v_k})$. The prune step discards all $(k+1)$-dimensional hyper-rectangles that have a k-dimensional projection outside $\mathcal{R}_{\geq ms}^{(k)}$. Again, the correctness of this procedure follows from the downward closure property of mass.

Choice of (ms, μ)**:** To obtain hyper-rectangles of varying size and quality, packing potentially different anomalies (and non-anomalies), we run Algorithm 1 with "conservative" parameters, i.e., low ms and high μ. As such, to generate

a good volume of candidates, we set (ms, μ) as the median of the number of anomalous points, normal points from the 1-dimensional hyper rectangles. Setting a higher ms and lower μ would prune more (and potentially undesirably many) candidates in exchange of reduced time. We use the median to strike a balance between the quality and running time. As we describe later in §3.3, all these candidate packs are forwarded to a selection algorithm, which carefully chooses the subset that yields the shortest description of all the anomalies. As such, even though there are parameters input to Algorithm 1, we do not expect them from the user, rather we vary and set those so as to generate various candidate packs. Having more candidates is likely to increase our chance of finding a combination that explains the anomalies the best (i.e., fewest bits).

To conclude, we note that other subgroup discovery techniques designed to handle numerical attributes, such as SubgroupMiner [26] or MIDOS [57], can be used as an alternative to Algo. 1; provided necessary modifications are incorporated to enforce the purity criterion.

3.2 Refining Hyper-rectangles into Hyper-ellipsoids

Grid or interval-based subspace clustering algorithms are limited to finding box-shaped rectangular clusters, and they may miss clusters inadequately oriented or shaped. To allow more flexibility, we refine each hyper-rectangle found by SUB-CLUS into a hyper-ellipsoid (which we call a *pack*, recall Definition 1). An ellipsoid with center $\mathbf{c}$ is written as

$$p(\mathbf{c}, \mathbf{M}) = \{\mathbf{x}|\ (\mathbf{x} - \mathbf{c})^T \mathbf{M}^{-1}(\mathbf{x} - \mathbf{c}) \leq 1\}$$

for positive semi-definite matrix $\mathbf{M} \succ 0$.

Given a hyper-rectangle R, let us denote the anomalous points it contains by $\mathbf{x}_i \in \mathcal{A}$ for $i = 1, \ldots, a_R$ (See Def.n 3) and anomalous points outside R by $\mathbf{x}_j \in \mathcal{A}$ for $j = a_{R+1}, \ldots, a$. The normal points are denoted by $\mathbf{x}_l \in \mathcal{N}$ for $l = 1, \ldots, n$.

When we convert a given R to an ellipsoid, we would like all $\mathbf{x}_i$'s (anomalous points) it already contains to reside inside the ellipsoid. In contrast, we would like all $\mathbf{x}_l$'s (normal points) to remain outside the ellipsoid. The refinement is achieved by enclosing as many as the other anomalous points ($\mathbf{x}_j$'s) that are in the vicinity of R inside the ellipsoid as well. Those would be the points that were left out due to axis-aligned interval-based box shapes that hyper-rectangles are limited to capture. An illustration is given in Fig. 3.

First we describe our approach for $\mathbf{x}_i$'s and $\mathbf{x}_l$'s, the positive and negative points that we respectively aim to include and exclude. The goal is to find a discriminating function $h(\cdot)$ where $h(\mathbf{x}_i) > 0$ and $h(\mathbf{x}_l) < 0$. To this end, we use the quadratic function $h(\mathbf{x}) = \mathbf{x}^T \mathbf{U}\mathbf{x} + \mathbf{w}^T \mathbf{x} + w_0$, with parameters $\boldsymbol{\Theta} = \{\mathbf{U}, \mathbf{w}, w_0\}$. We solve for $\boldsymbol{\Theta}$ by setting up an optimization problem based on a semi-definite program (SDP), that satisfies $\mathbf{x}_i^T \mathbf{U}\mathbf{x}_i + \mathbf{w}^T \mathbf{x}_i + w_0 > 0$ for all i and $\mathbf{x}_l^T \mathbf{U}\mathbf{x}_l + \mathbf{w}^T \mathbf{x}_l + w_0 < 0$ for all l. Most SDP solvers do not work well with strict inequalities, thus we modify to a non-strict feasibility problem by adding a margin, and solve (for each R):
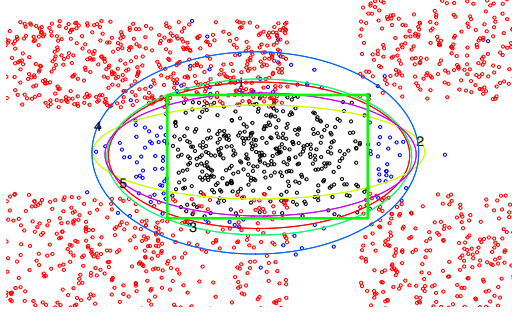
Fig. 3: Example illustration of refining hyper-rectangles to ellipsoids in 2-d. Anomalous points (black) captured by SubClus (Alg. 1) in a (green) rectangle, other anomalous points (blue) in the vicinity, and normal points (red).

$$\min_{\mathbf{U},\mathbf{w},w_0,\boldsymbol{\varepsilon}} \quad \sum \varepsilon_i + \lambda \sum \varepsilon_l$$
$$s.t. \quad \mathbf{x}_i^T\mathbf{U}\mathbf{x}_i + \mathbf{w}^T\mathbf{x}_i + w_0 \geq 1 - \varepsilon_i, \quad i = 1,\dots,a_R$$
$$\mathbf{x}_l^T\mathbf{U}\mathbf{x}_l + \mathbf{w}^T\mathbf{x}_l + w_0 \leq -1 + \varepsilon_l, \quad l = 1,\dots,n$$
$$\mathbf{U} \preceq -I, \quad \varepsilon_i \geq 0, \quad \varepsilon_l \geq 0$$

where $\mathbf{U}$ is a negative semi-definite matrix. We can show that $(\mathbf{U},\mathbf{w},w_0)$ define an ellipsoidal enclosing boundary, wrapping $\mathbf{x}_i$'s inside and leaving $\mathbf{x}_l$'s outside, for which we allow some slack ε. λ is to account for the imbalance between the number of positive and negative samples. The optimization problem is convex, which we solve using an efficient off-the-shelf solver, where each hyper-rectangle output by SubClus can be processed independently.

Having set up our refinement step as a convex quadratic discrimination problem, we next describe how we incorporate $\mathbf{x}_j$'s (anomalous points outside R) into the optimization. Intuitively, we would like to include as many other anomalies as possible inside the ellipsoid, but only those that are nearby $\mathbf{x}_i$'s and not necessarily those that are far away. In other words, we only want to "recover" the $\mathbf{x}_j$'s surrounding a given R and not grow the ellipsoid to include far away $\mathbf{x}_j$'s to the extent that it would end up including many normal points as well.

To this end, we treat $\mathbf{x}_j$'s similar to $\mathbf{x}_i$'s but incur a lower penalty of excluding an $\mathbf{x}_j$ than excluding an $\mathbf{x}_i$ or including an $\mathbf{x}_l$. The optimization is re-written as

$$\min_{\mathbf{U},\mathbf{w},w_0,\boldsymbol{\varepsilon}} \quad \sum \varepsilon_i + \alpha \sum \varepsilon_j + \lambda \sum \varepsilon_l$$
$$s.t. \quad \mathbf{x}_i^T\mathbf{U}\mathbf{x}_i + \mathbf{w}^T\mathbf{x}_i + w_0 \geq 1 - \varepsilon_i, \quad i = 1,\dots,a_R$$
$$\mathbf{x}_j^T\mathbf{U}\mathbf{x}_j + \mathbf{w}^T\mathbf{x}_j + w_0 \geq 1 - \varepsilon_j, \quad j = a_{R+1},\dots,a$$
$$\mathbf{x}_l^T\mathbf{U}\mathbf{x}_l + \mathbf{w}^T\mathbf{x}_l + w_0 \leq -1 + \varepsilon_l, \quad l = 1,\dots,n$$
$$\mathbf{U} \preceq -I, \quad \varepsilon_i \geq 0, \quad \varepsilon_j \geq 0, \quad \varepsilon_l \geq 0$$

Here, setting α (penalty constant for $\mathbf{x}_j$'s) smaller than both 1 and λ is likely a good choice. However, we do not know which (α,λ) pair would provide a good

trade-off in general. Therefore, we sweep over a grid of possible values[5] and generate various ellipsoids, as illustrated for the example case in Fig. 3. A last but important step is to sweep over the collection to discard *dominated* packs. Specifically, we output only the set of p's in the Pareto frontier w.r.t. mass versus purity. In this set there are *no two packs where one strictly dominates the other*—by enclosing *both* higher number of anomalous points (higher mass) and lower number of normal points (higher purity).

We refine a hyper-rectangle $R = (s_1, s_2, \ldots, s_{d'})$ into an ellipsoid within the same subspace, in other words, $\mathbf{U} \in \mathbb{R}^{d' \times d'}$ and $\mathbf{w} \in \mathbb{R}^{d'}$. For interpretability, we constrain $\mathbf{U}$ to be diagonal to obtain *axis-aligned* ellipsoids as shown in Fig. 3, since the original features have meaning to the user.[6]

Our explanation consists of one rule on each feature in the subspace. A feature rule is a $\pm$ radius interval around the ellipsoid's center. Formally:

Definition 4 (Feature rules) Given an axis-aligned ellipsoid $p(\mathbf{c}, \mathbf{M})$ in a subspace $f_{t_1} \times \ldots \times f_{t_{d'}}$, a rule on feature t_z is an interval $(\mathbf{c}[z] - radius_z, \mathbf{c}[z] + radius_z)$, where $radius_z = \sqrt{\mathbf{M}_{zz}}$, $\forall z = \{1, \ldots, d'\}$. Conjunction of all d' feature rules constitute the *signature* of p.

To wrap up, we show how to compute $\mathbf{c}$ and $\mathbf{M}^{-1}$ from $(\mathbf{U}, \mathbf{w}, w_0)$ to obtain the center and radii for an ellipsoid, using which we generate the feature rules. *Obtaining $\mathbf{c}$:* At the boundary of the ellipsoid, $h(\mathbf{x}) = 0$ and inside $h(\mathbf{x}) > 0$. Center is the point where $h(\mathbf{x})$ is the maximum. Hence;

$$\mathbf{c} := \max_{\mathbf{x}} \ \mathbf{x}^T \mathbf{U} \mathbf{x} + \mathbf{w}^T \mathbf{x} + w_0 = -\frac{1}{2} \mathbf{U}^{-1} \mathbf{w} \qquad (1)$$

Obtaining $\mathbf{M}^{-1}$:

$$\mathbf{x}^T(-\mathbf{U})\mathbf{x} - \mathbf{w}^T\mathbf{x} - w_0 < 0 \qquad (2)$$
$$\mathbf{x}^T(-\mathbf{U})\mathbf{x} + 2\mathbf{c}^T\mathbf{U}\mathbf{x} - w_0 < 0 \quad \text{using Eq. (1)}$$
$$(\mathbf{x} - \mathbf{c})^T(-\mathbf{U})(\mathbf{x} - \mathbf{c}) + \mathbf{c}^T\mathbf{U}\mathbf{c} - w_0 < 0$$
$$(\mathbf{x} - \mathbf{c})^T \frac{-\mathbf{U}}{(w_0 - \mathbf{c}^T\mathbf{U}\mathbf{c})} (\mathbf{x} - \mathbf{c}) < 1 \implies \mathbf{M}^{-1} = \frac{-\mathbf{U}}{(w_0 - \mathbf{c}^T\mathbf{U}\mathbf{c})}$$

Obtaining radii:

$$(\mathbf{x} - \mathbf{c})^T\mathbf{M}^{-1}(\mathbf{x} - \mathbf{c}) = \sum_{z=1}^{d'} (\mathbf{x}[z] - \mathbf{c}[z])^2 (\mathbf{M}^{-1})_{zz} \leq 1$$

To compute radius in dimension z, we find point $\mathbf{x}$ where $\mathbf{x}[z'] = \mathbf{c}[z']$, $\forall z' \neq z$, and $(\mathbf{x}[z] - \mathbf{c}[z])^2 \frac{1}{\mathbf{M}_{zz}} = 1$. It is easy to see that $radius_z = |\mathbf{x}[z] - \mathbf{c}[z]| = \sqrt{\mathbf{M}_{zz}}$.

---

[5] We use $\alpha = \{10^{-6}, 10^{-5}, \ldots, 1\} \times \lambda = \{10^{-3}, 10^{-2}, \ldots, 10^3\}$.

[6] If the anomalous patterns are to be used for detection, we estimate a full $\mathbf{U}$ matrix (i.e., possibly rotated ellipsoid).

3.3 Summarization: Pack Selection for Shortest Description

Our ultimate goal is to find anomalous patterns that explain or summarize the given anomalies in the dataset as succinctly as possible. Intuitively, "good" patterns enclose similar groups of points and hence help compress the data. To this end, we formulate our summarization objective by an encoding scheme and then devise an algorithm that carefully chooses a few patterns, in particular *packs* produced in §3.2, that yield the minimum encoding length. In the following, we describe our encoding scheme, followed by the proposed subset selection algorithm.

3.3.1 **MDL formulation for encoding a <u>given</u> *packing***

Our encoding scheme involves a Sender (us) and a Receiver (remote). We assume both of them have access to dataset $\mathcal{D} \in \mathbb{R}^{m \times d}$ but only the Sender knows the set of anomalous points $\mathcal{A}$. The goal of the Sender is to transmit (over a channel) to the Receiver the information about which points are the anomalies *using as few bits as possible*. Naïvely encoding all feature values of every anomalous point *individually* would cost $|\mathcal{A}| d \log_2 f$ bits.[7] The idea is that by encoding the enclosing boundary of *packs* (ellipsoids) found in §3.2, we (the Sender) could have the Receiver identify the anomalies *in groups*, which could save bits.

Obviously we would want to avoid "noisy" *packs* that include many normal points—that would necessitate spending extra bits for encoding those exceptions (i.e. "telling" the Receiver which points in a pack are *not* anomalies). Moreover, we would want to avoid using *packs* that encode largely overlapping group of anomalies, as bits would be wasted to redundancy. While identifying the *packing* that yields the fewest bits is the main problem, we first lay out our description length objective, for a <u>given</u> *packing* $\mathcal{P} = \{p_1(\mathbf{c}_1, \mathbf{M}_1), \ldots, p_K(\mathbf{c}_K, \mathbf{M}_K)\}$:

– Transmit number of *packs* $= \log^\star K$[8]

– For each pack $p_k \in \mathcal{P}$:

  – Transmit number of dimensions $= \log^\star d_k$, $d_k \leq d$

  – Transmit *identity* of dimensions $= \log_2 \binom{d}{d_k}$

  – Transmit the center $\mathbf{c}_k = d_k \log_2 f$

  – Transmit $\mathbf{M}_k = d_k^2 \log_2 f$ ($d_k \log_2 f$ if diagonal)

  – Transmit <u>exceptions</u> (i.e., non-anomalies in p_k):
    • number of normal points in $p_k = \log^\star n_k$

    • *identity* of normal points; by forming all possible subsets of size n_k of m_k (total number of points in p_k) $= \log_2 \binom{m_k}{n_k}$ (based on a canonical ordering of subsets, where points are ordered by distance to center)[9]

---

[7] Value of f is chosen according to the required floating point precision in the normalized feature space $\mathbb{R}^d$.

[8] Cost of encoding an arbitrary integer K is $L_{\mathbb{N}}(K) = \log^\star(K) + \log_2(c)$, where $c \approx 2.865064$ and $\log^\star(K) = \log_2(K) + \log_2(\log_2(K)) + \ldots$ summing only the positive terms [52]. We drop $\log_2(c)$ as it is constant for all *packing*s.

[9] Another way to identify the normal points in a *pack*: sort points by their distance to center and send the index of normal points in this list of length m_k. This costs more for $n_k \geq 2$: $n_k \log_2 m_k > \log_2 \frac{m_k^{n_k}}{n_k!} > \log_2 \binom{m_k}{n_k}$.

Total cost of encoding with *packing* $\mathcal{P}$ is then

$$\ell(\mathcal{P}) = \log^\star K + \sum_{k=1}^{K} L(p_k), \quad \text{where} \tag{3}$$

$$L(p_k) = \log^\star d_k + \log_2 \binom{d}{d_k} + d_k(d_k+1)\log_2 f + \log^\star n_k + \log_2 \binom{m_k}{n_k} \tag{4}$$

3.3.2 **MDL objective function**

Our objective is to find a *packing*, that is to identify a subset of *packs*, which provides the minimum encoding length. However, we do not assume that all anomalies would be covered by a packing, i.e., $\bigcup_k \mathcal{A}_k \subseteq \mathcal{A}$, as there could be anomalous points (outliers) that do not belong in any pattern but lie away from the others. The outliers $\mathcal{A}\backslash\{\bigcup_k \mathcal{A}_k\}$ are yet to be encoded individually.
Description length of all anomalies $\mathcal{A}$ with *packing* $\mathcal{P}$ is

$$L(\mathcal{A}|\mathcal{D},\mathcal{P}) = \big(|\mathcal{A}| - |\bigcup_{p \in \mathcal{P}} \mathcal{A}_p|\big)d\log_2 f + \big[\log^\star |\mathcal{P}| + \sum_{p \in \mathcal{P}} L(p)\big]$$

where the second term [in brackets] is $\ell(\mathcal{P})$: cost of transmitting $\mathcal{P}$ (and the anomalies covered by it) by Eq. (3), and the first term is the cost of individually encoding the remaining anomalies not covered by $\mathcal{P}$.

Notice that the objective of finding a subset $\mathcal{S}$ that minimizes the description length is equivalent to selecting a *packing* that reduces the naïve encoding cost of $|\mathcal{A}|d\log_2 f$ the most, i.e.:

$$\boxed{\max_{\mathcal{S}} \; R_\ell(\mathcal{S}) \; = \; |\bigcup_{p \in \mathcal{S}} \mathcal{A}_p|c_u - \log^\star |\mathcal{S}| - \sum_{p \in \mathcal{S}} L(p) + \big[\log^\star |\mathcal{E}| + \sum_{p' \in \mathcal{E}} L(p')\big]} \tag{5}$$

where $c_u = d\log_2 f$ is a constant unit-cost to encode a point, and set $\mathcal{E}$ denotes all the ellipsoids returned from the second part (refinement), as such, $\mathcal{S} \subseteq \mathcal{E}$. First three terms of the objective capture the overall *reduction* in encoding cost due to the packing with ellipsoids in $\mathcal{S}$. We can read it as aiming to *find a packing that covers as many anomalies as possible (expressive), while having small model cost (low complexity)—containing only a few packs in low dimensions.* The constant term [in brackets] ensures that $R_\ell(\mathcal{S})$ is a non-negative function.

3.3.3 **Subset selection algorithm for MDL *packing***

To devise a subset selection algorithm, we start by studying the properties of our objective function R_ℓ, such as submodularity and monotonicity that could enable us to use fast heuristics with approximation guarantees. Unfortunately, R_ℓ is not submodular as it is given in Eq. (5). However, with a slight modification where we fix the solution size (number of output *packs*) to $|\mathcal{S}| = K$, such that the second term is constant $\log^\star K$, the function becomes submodular, as we show below.

Theorem 1 *Our cardinality-constrained objective set function $R'_\ell(\mathcal{S})$ is submodular. That is, for all subsets $\mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{E}$ and packs $p \in \mathcal{E} \backslash \mathcal{T}$, it holds that*

$$R'_\ell(\mathcal{S} \cup \{p\}) - R'_\ell(\mathcal{S}) \geq R'_\ell(\mathcal{T} \cup \{p\}) - R'_\ell(\mathcal{T}) \;.$$

Proof Let $Cover(\mathcal{S}) = |\bigcup_{p \in \mathcal{S}} \mathcal{A}_p|$ return the number of anomalies contained by the union of *packs* in $\mathcal{S}$. Canceling the equivalent terms and constants on each side of the inequality, we are left with $Cover(\mathcal{S} \cup \{p\}) - Cover(\mathcal{S}) \geq Cover(\mathcal{T} \cup \{p\}) - Cover(\mathcal{T})$. The inequality follows from the submodularity property of the *Cover* function. $\square$

It is also easy to see that R'_ℓ is not monotonic.

Theorem 2 *Our modified objective set function $R'_\ell(\mathcal{S})$ is non-monotonic. That is, there exists $\exists \mathcal{S} \subseteq \mathcal{T}$ where $R'_\ell(\mathcal{T}) < R'_\ell(\mathcal{S})$.*

Proof For $\mathcal{S} \subseteq \mathcal{T}$, $Cover(\mathcal{T}) \geq Cover(\mathcal{S})$ due to monotonicity of *Cover* function. On the other hand, description cost of *packs* in $\mathcal{T}$ is $\sum_{p \in \mathcal{T}} L(p) = \sum_{p' \in \mathcal{S}} L(p') + \sum_{p'' \in \mathcal{T} \backslash \mathcal{S}} L(p'')$ and hence is strictly greater than those of $\mathcal{S}$. As such, for two *packing*s $\mathcal{S} \subset \mathcal{T}$ with the *same* coverage, we would have $R'_\ell(\mathcal{T}) < R'_\ell(\mathcal{S})$.[10] $\square$

Maximizing a submodular function is NP-hard as it captures problems such as Max-Cut and Max k-cover [17]. Nevertheless the structure of submodular functions makes it possible to achieve non-trivial results. In particular, there exist approximation algorithms for *non-monotone submodular* functions that are *non-negative*, like our objective function R'_ℓ. In particular, one can achieve an approximation factor of 0.41 for the maximization of any non-negative non-monotone submodular function *without* constraints [17].

In our case, we need to solve our objective under the cardinality (i.e., subset size) constraint, where $|\mathcal{S}|$ is fixed to some K (since only then R_ℓ is submodular). To this end, we use the RANDOM-GREEDY algorithm by Buchbinder et al. [6], which provides the best known guarantee for the cardinality-constrained setting, with approximation factors in $[0.356, \frac{1}{2} - o(1)]$. The algorithm is quite simple; at each step of K iterations, it computes the marginal gain of adding a single *pack* $p \in \mathcal{E} \backslash \mathcal{S}$ to $\mathcal{S}$ and selects one among the top K highest-gain *packs* uniformly at random.

Choice of K: We identify K, the number of *packs* to describe the anomalies, automatically, best of which is unknown apriori. Concretely, we solve to obtain subset $\mathcal{S}_K^\star$ each time for a fixed $K = |\mathcal{S}_K^\star| = 1, 2, \ldots, a$, and return the solution with the largest objective value of $R_\ell(\mathcal{S}_K^\star) = R'_\ell(\mathcal{S}_K^\star) - \log^\star K$ in Eq. (5). This is analogous to model selection with regularization for increasing model size.

3.4 Overall Algorithm x-PACS

Algorithm 2 puts together all three components of x-PACS as described through §3.1–§3.3. We conclude this section with the computational complexity analysis.

---

[10] Intuitively, this is where R_ℓ drops when we add a new *pack* to $\mathcal{S}$ (with positive cost) that does not cover any new anomalies.

Algorithm 2 X-PACS $(\mathcal{A} \cup \mathcal{N})$: Explaining Anomalous Patterns

---

Input: dataset $\mathcal{D} = \mathcal{A} \cup \mathcal{N}$ with labeled anomalies
Output: set of anomalous patterns (represented as hyper-ellipsoids)
    $\mathcal{P} = \{p_1(\mathbf{c}_1, \mathbf{M}_1), \ldots, p_K(\mathbf{c}_K, \mathbf{M}_K)\}$
1: Set of hyper-rectangles $\mathcal{R} = \emptyset$
2: Obtain $\mathcal{R}^{(1)}$ (1-d intervals) by kernel density estimation, varying cut-off threshold in $q = \{80, 85, 90, 95\}$
3: $\widehat{f_a} :=$ distribution of number of anomalies across $\mathcal{R}^{(1)}$
4: $\widehat{f_n} :=$ distribution of number of normal points across $\mathcal{R}^{(1)}$
5: $\mathcal{R} := \text{SUBCLUS}(\mathcal{D}, ms = q(\widehat{f_a}, 50), \mu = q(\widehat{f_n}), 50))$ by Alg. 1 in §3.1
6: Set of hyper-ellipsoids $\mathcal{E} = \emptyset$
7: **for** $R \in \mathcal{R}$ **do**
8:     $\mathcal{E}_R = \emptyset$
9:     **for** $\alpha = \{10^{-6}, 10^{-5}, \ldots, 1\}$ **do**
10:         **for** $\lambda = \{10^{-3}, 10^{-2}, \ldots, 10^3\}$ **do**
11:             $\mathcal{E}_R := \mathcal{E}_R \cup$ solve optimization problem in §3.2 for (R, α, λ)
12:         **end for**
13:     **end for**
14:     $\mathcal{E} := \mathcal{E} \cup \text{ParetoFrontier}(\mathcal{E}_R)$
15: **end for**
16: **for** $K = 1, \ldots, |\mathcal{A}|$: select a subset $\mathcal{S}_K^* \subset \mathcal{E}$ of K *packs* using the cardinality-constrained RANDOM-GREEDY algorithm by Buchbinder et al. [6] to optimize the description length reduction objective $R_\ell(\cdot)$ in §3.3.
17: **return** $\mathcal{P} := \arg\max_{\mathcal{S}_K^*} R_\ell'(\mathcal{S}_K^*) - \log^* K$

Computational complexity: We analyze the complexity of each part separately. Main computation of §3.1 is the SUBCLUS algorithm. Preliminary KDE to create 1-d intervals is independently done per dimension in parallel, only on the anomalous points. We use a constant number of sampling locations, as such, KDE complexity is $O(a)$ where a is the number of anomalies. SUBCLUS then proceeds level-by-level and makes as many passes over the data as the number of levels. For a d' dimensional hyper-rectangle that meets the mass and purity criteria, all its $2^{d'}$ projections in any subset of the dimensions also meet the mass criterion (although may not be pure). As such, running time of SUBCLUS is exponential in the highest dimensionality of the hyper-rectangle that meets both criteria. Total time complexity of this step is $O(c^{d_{\max}} + md_{\max})$ for a constant c[11] that accounts for possibly multiple $d_{\max}$-dimensional hyper-rectangles and the smaller ones. The second term captures the passes over the data over $d_{\max}$ levels.

The main computation of §3.2 is solving the SDP optimization problem, for which we use the popular cvx SDPT3 solver that takes $O([d_{\max} + m]^3)$ for an axis-aligned ellipsoid (or diagonal $\mathbf{U}$) per iteration.[12] To speed up, we filter bulk of the points beyond a certain distance of a given hyper-rectangle, since its refined hyper-ellipsoid would mostly include/exclude points inside and nearby it. Filtering takes $O(m)$, after which we solve the SDP for a near-constant number of points. It is easy to show that finding the Pareto frontier set of non-dominating packs (line 14)—such that no pack that has strictly larger mass *and* smaller impurity exists—can be done through two passes over all alternative hyper-ellipsoids generated for different (α, λ). This procedure does not change the overall complexity but

---

[11] For instance, if we have t $d_{\max}$-dimensional hyper-rectangles, then the complexity would be $O(t2^{d_{\max}} + md_{\max})$, we could rewrite this as $O(c^{d_{\max}} + md_{\max})$

[12] In practice, the solver converges in 20-100 iterations.

is likely to yield a much smaller set of ellipsoids per rectangle. We refine each hyper-rectangle independently in parallel.

The main computation in the last part is the RANDOM-GREEDY algorithm, which makes K iterations for a given number of packs K. In each iteration, it makes a pass over the not-yet-selected hyper-ellipsoids, computes the marginal reduction in bits by selecting each, and picks randomly among the top K with the highest reduction. We use a size-K min-heap to maintain the top K as we make a pass over the packs. Worst case cost is $O(|\mathcal{E}| \log K)$, multiplied by K iterations. We run RANDOM-GREEDY for $K = 1, \ldots, a$, each of which is parallelized. Total complexity of §3.3 is $O(|\mathcal{E}| a \log a)$.

The number of ellipsoids, $|\mathcal{E}|$, is in the same order of the number of hyper-rectangles from §3.1, i.e., $O(c^{d_{\max}})$. Thus, the overall complexity can be written as $O(m d_{\max} + c^{d_{\max}} a \log a)$; linear on the number of data points m, near-linear on a, and exponential in the largest pack dimensionality $d_{\max}$.

4 Experiments

Through experiments on real-world datasets we answer the following questions. A quick reference to the UCI datasets used in our experiments is in Table 1. Last column gives % savings (in bits) in describing/encoding the anomalies by x-PACS.

Q1. **Effectiveness:** How accurate, interpretable, and succinct are our explanations? How do they compare to descriptions by Decision Trees?

Q2. **Detection performance:** Do our explanations generalize? Can they be used as signatures to detect future anomalies? To this end, we compare x-PACS to 7 different baselines.

Q3. **Scalability:** How does x-PACS's running time scale in terms of data size and dimensionality?

4.1 Effectiveness of Explanations

Our primary focus is anomaly *description* where we unearth interpretable characteristics for known anomalies. To this end, we present 6 case studies with ground truth, followed by quantitative comparison to decision trees.

4.1.1 Case Studies

Our **Image dataset** contains gray-scale headshot images of various people. We designate the majority wearing dark-color t-shirts as the normal samples. We create 3 versions containing different number of anomalous patterns, as we describe below. We compare x-PACS's findings to the ground truth.

Case I: ImagesI We label 8 images of people wearing sunglasses as anomalies as shown in (a) below, and combine them with the normal samples none of which has sunglasses. In this simple scenario x-PACS successfully identifies a single, 1-d pattern shown in (b), which packs all the 8 anomalies but no normal samples. Also shown at the bottom of (b) is the interval of values, that is the $\pm$ radius range

Table 1: Dataset statistics. x-PACS achieves significant savings (in bits) by explaining anomalies in groups.

Name	size m	dim. d	anom. a	%-savings
ImagesI	88	120	8	99.75
ImagesII	91	180	9	88.53
ImagesIII	110	180	12	99.51
DigitI	1371	16	228	99.83
DigitII	1266	16	211	99.72
BrCancer	683	9	239	93.74
Arrythmia	332	172	87	92.92
Wine	95	13	24	97.04
Yeast	592	8	129	98.04

around the *pack*'s center, for the corresponding dimension (the lower, the darker the pixel).

(a) anomalies   (b) x-PACS *packing*
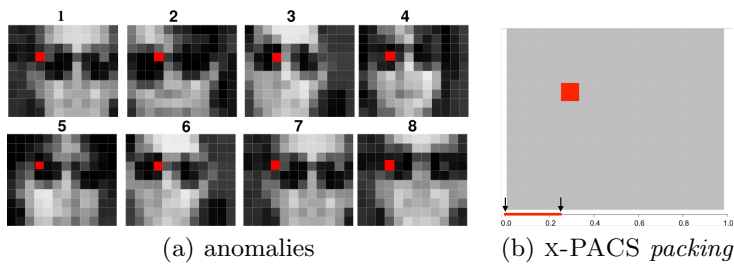
Case II: ImagesII Next, we construct the 9 anomalies as shown earlier in §2 in Fig. 1: 6 wearing sunglasses, 4 white t-shirt (2 wearing both), plus 1 person with a beard (normal samples has no beard). As detailed in the caption of the figure, x-PACS finds 2 pure *packs*, each 1-d, that collectively describe the 8 anomalies and none of the normal samples. The bearded image does not belong to any *pack* and is left out as an outlier.

Case III: ImagesIII We construct the third dataset with 12 anomalies: the same 9 from ImagesII plus 3 faces (10–12) with beard as shown below. In this case, x-PACS finds that characterizing the bearded images as a separate pattern is best to reduce the description cost, and outputs 3 pure, 1-d *packs* shown in (b).

(a) +3 anomalies   (b) x-PACS *packing*
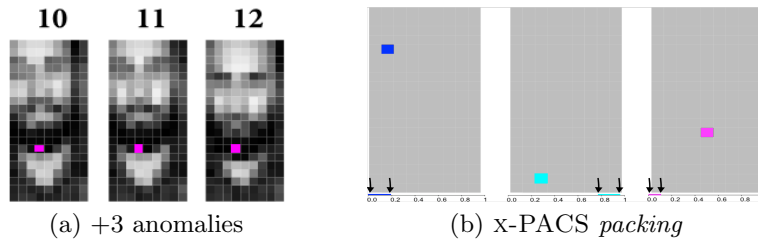
In all scenarios, x-PACS is able to unearth simple (low-dimensional) and pure (discriminative) characteristics of the anomalies. Also, it automatically identifies the correct number of anomalous patterns that yield the shortest data description as shown in Fig. 4.
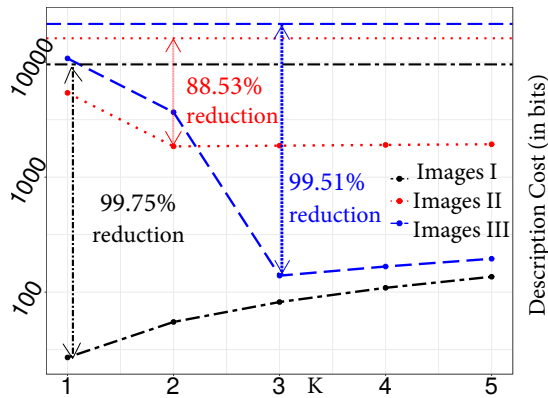
Fig. 4: x-PACS's description cost of anomalies in image datasets for $K = 1, \ldots, 5$. Naïve/base cost ($K = 0$) is shown w/ a horizontal line per dataset. x-PACS finds the appropriate number of patterns automatically and significantly reduces the description cost.

Next we study a different domain. The **Digit dataset** contains instances of digit hand-drawings in time. Features are the x and y coordinates of the hand in 8 consecutive time ticks during which a human draws each digit on paper. As such, each drawing has 16 features.

Case IV: DigitI We designate all drawings of digit '0' as normal and a sample of digit '7' as 'anomalous' to study the characteristics of drawing a '7' as compared to a '0'. 8 different positions of the hand in time averaged over all corresponding samples of these two digits is shown below (a–b).

| (a) avg. '0'-drawing | (b) avg. '7'-drawing | (c) x-PACS *packing* |
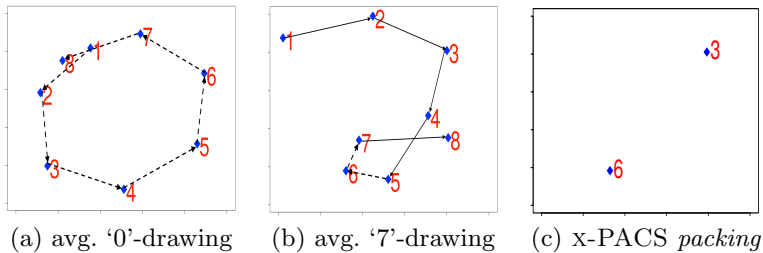
x-PACS identifies a single, 2-d *pack* containing all 228 instances of '7's and no '0's, as given in Table 2, where we list the ellipsoid center and the $\pm$radius interval where the hand is positioned for the characterizing features. The anomalous pattern suggests right & bottom positioning of the hand respectively at times t3 & t6, which follows human intuition—in contrast, typical hand positions for '0' at those ticks are opposite; at the left & top. Corresponding avg. hand positions in 2-d is shown in (c) above.

Table 2: DigitI '0' vs. '7': x-PACS finds one 2-d *pack*.

packID	feature	center	interval	$|\mathcal{A}_k|$	$|\mathcal{N}_k|$
$k = 1$	$x@t3$	0.82	(0.66, 0.98)	228	0
	$y@t6$	0.17	(0.02, 0.31)		

Case V: `DigitII` We perform a second case study where we designate digit '8' drawings as normal and '2' and '3' as the anomalies. Avg. drawings are illustrated in (a–b) below. x-PACS is able to describe 210 of the 211 anomalies in a single, 4-d pack listed in Table 3 and illustrated in (c). The single unpacked drawing is shown in (d) and looks like an odd '3'.
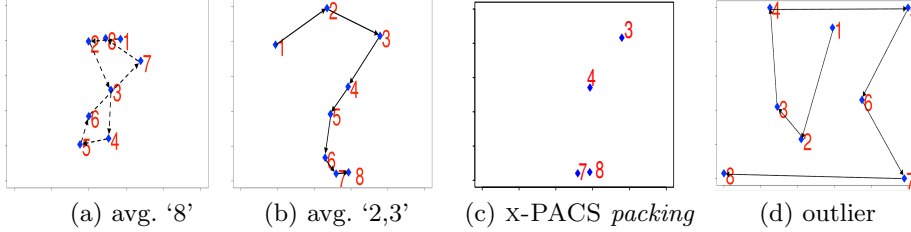

(a) avg. '8'   (b) avg. '2,3'   (c) x-PACS *packing*   (d) outlier

Table 3: `DigitII` '8' vs. '2','3': x-PACS's one 4-d *pack*.

packID	feature	center	interval	$|\mathcal{A}_k|$	$|\mathcal{N}_k|$
$k = 1$	$y@t3$	0.83	(0.71, 0.95)	210	0
	$y@t4$	0.54	(0.38, 0.69)		
	$y@t7$	0.04	(0.00, 0.11)		
	$y@t8$	0.05	(0.00, 0.12)		

Looking at the avg. '8' vs. '2' or '3' drawings above, it appears that a single feature like $y@t8$, i.e., vertical hand position at the end, should be discriminative alone; as '8' tends to end at the *top* vs. others at the bottom.

Interestingly, none of the 1-d *packs* on $y@t8$ is pure like the 4-d one output. A non-anomalous sample it contains is shown on the right, which is an '8' that starts and ends at the *bottom* just like most '2' and '3's.
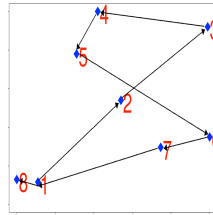
Case VI: `BrCancer` Finally, **breast cancer dataset** contains 239 malign (anomalous) and 444 benign cancer instances. x-PACS finds 5 *packs* listed in Table 4, covering a total of 226 anomalies while also including 17 unique normal points in the *packing*. *Pack* 1 characterizes 162 cases with high 'chromatin'. Second 2-d *pack* suggests large 'clumpthickness' and 'mitoses' (related to cell division and tissue growth) for 145 cases. Smaller pure 1-d *packs*, 4 and 5, indicate very large 'cellsize' and 'nucleoili'. These findings are intuitive even to non-experts like us (although we lack the domain expertise to interpret *pack* 3).

Table 4: `BrCancer`: x-PACS finds five 1-d or 2-d *packs*.

packID	feature	center	interval	$|\mathcal{A}_k|$	$|\mathcal{N}_k|$
$k = 1$	chromatin	0.76	(0.63, 0.88)	162	11
$k = 2$	clumpthickness	0.94	(0.84, 1.00)	145	5
	mitoses	0.28	(0.00, 0.63)		
$k = 3$	epicellsize	0.33	(0.24, 0.42)	97	2
	barenuclei	0.11	(0.09, 0.14)		
$k = 4$	nucleoili	0.98	(0.93, 1.00)	75	0
$k = 5$	cellsize	0.99	(0.98, 1.00)	67	0

Table 5: Interpretability measures (a)–(d): x-PACS vs. Rule learners. Also given for reference is detection performance in AUPRC (See §4.2 for details).

measure /method	(a) # of groups	(b) avg. length	(c) avg. impurity	(d) avg. width	detection performance
DT-5	4.0000	2.9889	0.0233	0.4769	0.6252
DT-4	3.7778	2.7856	0.0422	0.4801	0.6070
DT-3	3.0000	2.4078	0.0700	0.4812	0.6210
DT-2	2.4444	1.8889	0.1378	0.4872	0.6236
DT-1	**1.7778**	**1.0000**	0.4056	0.5017	0.5656
RuleFit[13]	12.0000	1.7800	0.0229	0.4643	0.8471
Ripper	2.4444	1.5244	0.0178	0.3889	0.7244
x-PACS	2.5556	2.1000	**0.0152**	**0.2333**	**0.8781**

4.1.2 x-PACS *vs. Rule Learners*

Since in our work, we view anomalies as an already defined class, explaining anomalies is equivalent to describing an under represented target class [57]. Hence, we compare x-PACS to techniques that explain labeled data. To this end, we consider interpretable supervised models, specifically, inductive rule based learners that aim to extract rules from a labeled data set that are discriminative in nature. We argue that linear classifiers like logistic regression are not comparable to x-PACS for two key reasons. First, they *do not group* the anomalies, but rather output a single separating hyperplane. Second, they *do not provide rules on the features*, but only feature coefficients, which could be negative (hard to interpret). Further, techniques aiming to explain black box predictions are not directly comparable to our method since most of the works aim to explain one instance at a time compared to the group wise explanations x-PACS provides.

We compare x-PACS to the following popular rule based learners.

1. Decision Tree (DT): DT aims to partition (or group) the labeled data into pure leaves. We treat the leaves containing at least two anomalies analogous to our *packs*. Each such leaf is characterized by the feature rules (or predicates) on the path from the root.
2. Ripper [9]: Ripper is a popular inductive rule learner that sequentially mines for feature rules with high accuracy and coverage with the aim to achieve generalization. We use a publicly available implementation of Ripper in the Weka repository for our experiments and consider rules that are labeled anomalous.
3. RuleFit [15]: RuleFit is an ensemble learner where the base learner is a rule generated by a decision tree. A regression/classification is setup using the base learners to identify the rules that are important in discriminating the different classes. We use the publicly available RuleFit[14] implementation and use the rules with non-zero coefficients with atleast two anomalies.

To compare x-PACS with rule learners, it is not fair to use description length since the listed techniques do not explicitly optimize it. Instead, we use the following external interpretability measures proposed in [35] (all being lower the better):

---

[13] Note that RuleFit is averaged over seven datasets due to underspecified regression in `Arrythmia` and `Yeast`

[14] R package pre : https://CRAN.R-project.org/package=pre

(a) # groups

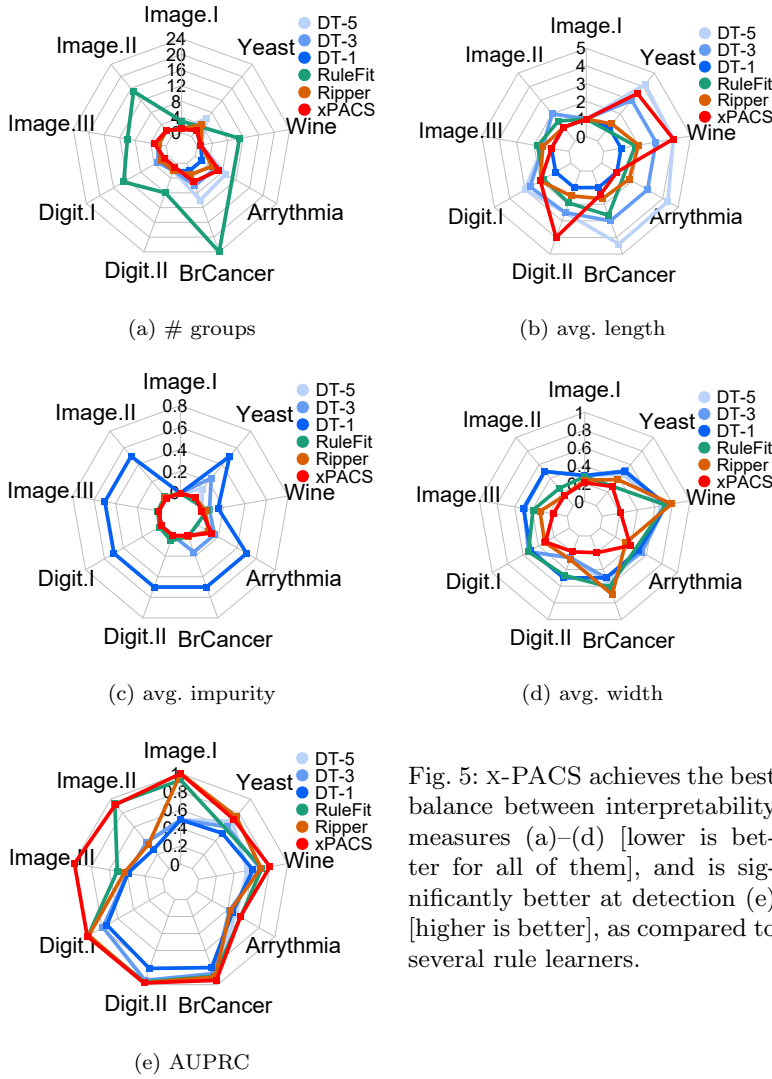(b) avg. length

(c) avg. impurity

(d) avg. width

(e) AUPRC

Fig. 5: x-PACS achieves the best balance between interpretability measures (a)–(d) [lower is better for all of them], and is significantly better at detection (e) [higher is better], as compared to several rule learners.

(a) number of groups (anomalous packs), (b) avg. length of rules (pack dimensionality), (c) avg. fraction of normal points within packs (impurity divided by n), and (d) avg. interval width across feature rules. In other words, an explanation with fewer groups, fewer rules, fewer exceptions, and smaller spread in features is considered more interpretable.

DT has no means to choose the number of packs automatically. Therefore, we report DT results for depths 1–5 as compared to x-PACS in Table 5, averaged across all datasets. In addition to the interpretability measures, we report the detection performance in AUPRC (area under precision-recall curve) on held-out data (80-20 split) that quantifies the generalization of the subspace rules. Results on individual datasets per measure are shown with radar charts in Fig. 5. (See Table 6 for detailed results.) Notice the trade-offs between the measures for DT:

while (c) and (d) tend to decrease with increasing depth, (a) and (b) increase. The lack of rule summarization in RuleFit is evident in the number of groups (a) where x-PACS consistently produces smaller number of explanations across various data sets. We also note that x-PACS produces tighter intervals (d) compared to Ripper emphasizing the concreteness of the explanations. Overall, x-PACS achieves the best trade-off with lower overall values across the interpretability measures. Moreover, our signatures are significantly better at detecting future anomalies. We present more detailed experiments on detection next.

Table 6: Rule learners and DT (with respective depths 1–5) compared to x-PACS across datasets on interpretability measures (a)–(d) [all lower the better] as well as detection performance AUPRC [higher the better]. RuleFit leads to underspecified regression in `Arrythmia` and `Yeast` which we denote by NA.

Measure	Dataset/ Model	Image I	Image II	Image III	Digit I	Digit II	Br Cancer	Arry thmia	Wine	Yeast
(a) number of groups	DT-5	1.00	2.00	2.00	3.00	2.00	10.00	9.00	1.00	6.00
	DT-4	1.00	2.00	2.00	3.00	2.00	8.00	9.00	1.00	6.00
	DT-3	1.00	2.00	2.00	3.00	2.00	6.00	6.00	1.00	4.00
	DT-2	1.00	2.00	2.00	3.00	2.00	4.00	4.00	1.00	3.00
	DT-1	1.00	2.00	2.00	2.00	2.00	2.00	2.00	1.00	2.00
	RuleFit	3.00	15.00	10.00	13.00	8.00	24.00	NA	11.00	NA
	Ripper	1.00	2.00	2.00	2.00	2.00	3.00	5.00	1.00	4.00
	x-PACS	1.00	2.00	3.00	1.00	1.00	5.00	7.00	1.00	2.00
(b) avg. length of rules	DT-5	1.00	2.00	1.50	3.00	2.50	4.40	4.33	4.00	4.17
	DT-4	1.00	2.00	1.50	3.00	2.50	3.63	3.78	4.00	3.67
	DT-3	1.00	2.00	1.50	2.67	2.50	3.00	3.00	3.00	3.00
	DT-2	1.00	2.00	1.50	2.00	2.50	2.00	2.00	2.00	2.00
	DT-1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	RuleFit	1.00	1.40	1.80	1.80	1.92	2.67	NA	1.81	NA
	Ripper	1.00	1.00	1.50	2.00	1.50	1.67	1.80	2.00	1.25
	x-PACS	1.00	1.00	1.00	2.00	4.00	1.40	1.00	4.00	3.50
(c) avg. fraction of normal points	DT-5	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.11
	DT-4	0.00	0.00	0.00	0.00	0.00	0.11	0.11	0.00	0.16
	DT-3	0.00	0.00	0.00	0.00	0.00	0.16	0.16	0.07	0.24
	DT-2	0.00	0.00	0.00	0.32	0.01	0.25	0.25	0.08	0.33
	DT-1	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.15	0.50
	RuleFit	0.00	0.02	0.01	0.02	0.05	0.01	NA	0.05	NA
	Ripper	0.00	0.01	0.00	0.00	0.00	0.01	0.10	0.03	0.01
	x-PACS	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.01
(d) avg. interval width	DT-5	0.29	0.50	0.49	0.53	0.26	0.56	0.53	0.68	0.46
	DT-4	0.29	0.50	0.49	0.53	0.26	0.54	0.53	0.68	0.51
	DT-3	0.29	0.50	0.49	0.53	0.26	0.50	0.54	0.72	0.51
	DT-2	0.29	0.50	0.49	0.51	0.29	0.50	0.50	0.79	0.52
	DT-1	0.29	0.50	0.50	0.50	0.50	0.50	0.50	0.73	0.50
	RuleFit	0.28	0.25	0.39	0.53	0.47	0.61	NA	0.72	NA
	Ripper	0.24	0.15	0.31	0.32	0.28	0.7	0.33	0.79	0.38
	x-PACS	0.21	0.15	0.16	0.31	0.19	0.20	0.39	0.21	0.28
AUPRC	DT-5	0.49	0.38	0.39	0.79	0.95	0.89	0.50	0.55	0.68
	DT-4	0.49	0.38	0.39	0.79	0.95	0.86	0.50	0.55	0.55
	DT-3	0.49	0.38	0.39	0.79	0.95	0.87	0.42	0.69	0.60
	DT-2	0.49	0.38	0.39	0.79	0.95	0.86	0.47	0.61	0.67
	DT-1	0.49	0.27	0.39	0.74	0.81	0.80	0.46	0.61	0.51
	RuleFit	0.93	0.93	0.51	0.98	0.97	0.90	NA	0.71	NA
	Ripper	1.00	0.35	0.43	0.97	0.98	0.91	0.43	0.70	0.75
	x-PACS	1.00	0.92	0.99	0.99	0.98	0.95	0.56	0.80	0.71

4.1.3 Ablation Study

We study the importance of the refinement step discussed in §3.2 by performing an ablation study. To this end, we omit the refinement of hyper-rectangles into hyper-ellipsoids in x-PACS (denoted as ablated x-PACS). Recall that the primary reason we perform the refinement step is to cover more anomalous points and reduce the number of normal points in the packs (See Fig. 3). Hence, to showcase the benefit, in Table 7, we compare the proportion of anomalous (higher is better) and normal points (lower is better) covered in the final packs obtained using x-PACS and the ablated x-PACS. In addition, we also report the %-savings (higher is better) achieved in both cases. In x-PACS, the summarization step (See §3.3) transmits the center and the diagonal matrix of the packs §3.3.1. To accommodate hyper rectangles, we modify this to transmit the upper and lower bounds of the hyper rectangles in the ablated x-PACS.

Table 7: Ablation Study: x-PACS vs. ablated x-PACS (no refinement to ellipsoids). Coverage of anomalous points (higher is better), coverage of normal points (lower is better), and % savings (higher is better).

	Method	Images I	Images II	Images III	Digit I	Digit II	Br Cancer	Arry thmia	Wine	Yeast
Coverage of anom. points	x-PACS	**1.00**	**0.89**	**1.00**	**1.00**	**1.00**	**0.95**	**0.93**	**0.96**	**0.68**
	ablated x-PACS	**1.00**	**0.89**	**1.00**	0.96	0.89	0.88	0.78	0.92	0.62
Coverage of normal points	x-PACS	**0**	**0**	**0**	**0**	**0**	**0.03**	**0.35**	**0.11**	**0.10**
	ablated x-PACS	**0**	**0**	**0**	0.01	0.01	0.05	0.53	0.18	0.13
%-savings	x-PACS	**99.75**	**88.53**	**99.51**	**99.83**	**99.72**	**93.74**	**92.92**	**97.04**	**98.04**
	ablated x-PACS	**99.75**	**88.53**	**99.51**	92.11	87.21	85.68	78.16	91.42	90.51

From Table 7, we observe that x-PACS is indeed able to cover more anomalous points, while avoiding normal points in the final packs for all the datasets. These results demonstrate the utility of the refinement step.

4.2 Detection Performance

While not our primary focus, x-PACS can also be used to *detect* anomalies. Specifically, given the *packs* identified from historical/training data, a future test instance that falls in any one of the *packs* (i.e., enclosed within any hyper-ellipsoid in the *packing*) can be flagged as an anomaly.[15]

To measure detection quality, we compare x-PACS to 7 competitive baselines on all datasets.

1. Mixture of K-GAUSSIANS on the anomalous points. $K \in \{1, 2, \ldots, 9\}$ chosen at the "knee" of likelihood. Anomaly score of test instance: maximum of the probabilities of being generated from each cluster.

2. KDE on the normal points. Gaussian kernel bandwidth chosen by cross-validation. Anomaly score: negative of the density at test point.

---

[15] Note that, like any supervised method, x-PACS could only detect future instances of anomalies of known types.

Table 8: Area under precision-recall curve (AUPRC) on anomaly detection.

Method	ImagesI	ImagesII	ImagesIII	DigitI	DigitII	BrCancer	Arrythmia	Wine	Yeast
K-GAUSSIANS	0.182	0.239	0.184	0.162	0.333	0.613	0.227	0.258	0.265
KDE	0.952	0.978	0.987	0.989	0.997	0.981	0.571	0.667	0.681
NN	0.491	0.472	0.659	0.967	0.821	0.520	0.546	0.562	0.348
PCA+SVDD	0.286	0.217	0.212	0.331	0.529	0.861	0.295	0.566	0.606
DT	0.802	0.764	0.812	0.831	0.961	0.884	0.516	0.637	0.673
SVM-LIN	**1.000**	**1.000**	**1.000**	0.999	0.999	**0.984**	0.755	**0.994**	0.823
SVM-RBF	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.964	**0.810**	0.984	**0.861**
x-PACS	**1.000**	0.921	0.990	0.993	0.976	0.951	0.564	0.799	0.701

3. NN. Anomaly score: distance of test point to its nearest neighbor (nn) normal point in training set, divided by the distance of that nn point to its own nearest normal point in training set.

4. PCA+SVDD on all points [55]. A *single* hyperball that aims to enclose anomalous points in the *PCA-reduced* space[16], for which the embedding dimensionality is chosen at the "knee" of the scree plot. Anomaly score: distance of test point from the hyperball's center.

5. DT on all points, where we balance the data for training and regularize by tree-depth, chosen from $\{1, 2, \ldots, 30\}$ via cross-validation. Anomaly score: number of anomalous samples in the leaf the test point falls into divided by leaf size.

6-7) SVM-LIN & SVM-RBF on all points. Hyperparameters set by cross-validation. Anomaly score: "confidence", i.e., distance from decision boundary.

We create 3 folds of each dataset, and in turn use 2/3 for training and 1/3 for testing, except the `Images` datasets with the fewest anomalies for which we do leave-one-out testing. All points receive an anomaly score by each method as described above. x-PACS's anomaly score for a test instance $\mathbf{x}$ is the maximum $h_k(\mathbf{x}) = \mathbf{x}^T \mathbf{U}_k \mathbf{x} + \mathbf{w}_k^T \mathbf{x} + w_{0k}$ among all p_k's in the *packing* resulting from training data. We rank points in decreasing order of their score, and report the area under the precision-recall curve in Table 8.

SVMs achieve the highest detection rate, as might be expected. However, kernel SVM cannot be interpreted. Linear SVM, like LR, does not identify anomalous patterns nor does it produce any explicit feature rules. Notably, x-PACS outperforms all other baselines considerably across datasets, including DT, which produces the most interpretable output among the baselines as discussed in §4.1.2.

4.3 Scalability

Finally, we quantify the scalability of x-PACS empirically. To this end, we implement a synthetic data generator, parameterized by data size, total dimensionality, maximum pack size and dimensionality and number of anomalous packs. Anomalies are sampled from a small range per feature within a subspace, and normal

---

[16] SVDD optimization diverged for some high dimensional datasets, therefore, we performed PCA as a preprocessing step.

points are sampled from the reverse of the histogram densities derived from the anomalous points.

Fig. 6 shows the running time w.r.t. data size m, dimensionality d, average pack dimensionality d_{avg}, and total number of anomalies a. All plots demonstrate near-linear scalability. Recall that we showed exponential complexity w.r.t. maximum pack dimensionality $d_{\max}$. Notably, we observe linear time growth on average.
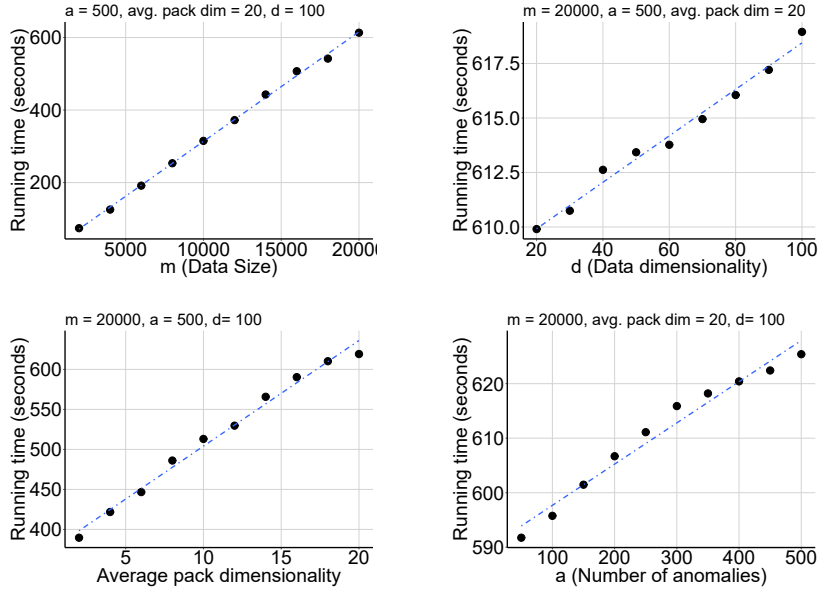

Fig. 6: x-PACS scales linearly with input size.

5 Related Work

Related areas of study span across outlier explanation, subspace clustering and subspace outlier detection, data description, subgroup discovery, rule learning, rare class discovery and methods to explain black-box classifiers. We illustrate related work in the context of our desiderata in Table 9.

Outlier explanation: The seminal work by Knorr and Ng [27] provides what they call "intensional knowledge", per outlier, by identifying the minimal subspaces in which it deviates. To find the optimal subset of features that differentiate the outliers from normal points, [34] formulates a constraint programming problem and [24] takes a subspace search route. Similarly, [10, 11, 40] aim to explain one outlier at a time by features that participate in projection directions that maximally separate them from normal points. *All* existing work in this area assume the outliers are *scattered* and strive to explain them *individually* rather than in groups. Therefore, they cannot identify anomalous patterns. Moreover, they do not focus explicitly on shortest description, let alone in a principled, information-theoretic way as we address in this work.

Table 9: Comparison of related work in terms of properties D1–D5 in reference to our Desiderata (see §1.1).

Property	Explain as-a-group?	Multiple groups?	Find subspace?	Rules on features?	Discriminative?	Minimal?
Subspace clustering [3,7,53,30,44]	✓	✓	✓	✓		
Projected clustering [2,41]	✓	✓	✓			
(data descr.) SVDD [55], SSSVDD [18]	✓			✓	✓	
(rare category) RACH [21]	✓			✓	✓	
(rare category) PALM [20]	✓		✓	✓	✓	
Knorr and Ng [27]			✓		✓	✓
RefOUT [24], CP [34] LODI [11], LOGP [10]			✓		✓	
EXPREX [5]	✓		✓	✓	✓	
(Explaining black box classifiers) LIME [51]			✓		✓	
EXstream [58]	✓		✓	✓	✓	✓
Explainer [50]			✓	✓	✓	
SRF [29], Krimp [56], RuleFit [15], Ripper [9]	✓	✓	✓	✓	✓	
x-PACS [this paper]	✓	✓	✓	✓	✓	✓

Extending earlier work [4] on explaining single outliers, [5] aims to explain groups of outlier points or what they call sub-populations. They search for ⟨context, feature⟩ pairs, where the (single) feature can differentiate as many outliers as possible from normal points that share the same context. It is important to note that their goal is to explain a group (or set) of outliers altogether and not particularly explaining them with multiple groups. Similarly, [58] 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. Again, they strive to explain all the anomalies as a group, and not in multiple groups.

We found that SRF (sapling random forest) [29] aims to explain and cluster outliers similar to our problem setting. They build on their earlier work [50], which explains outliers one at a time by learning an ensemble of small decision trees (called saplings) and combining the rules (from root to leaf in which the outlier lies) across the trees. SRF then groups the outliers using k-means clustering based on the similarity of their explanations. However, there is no guarantee on the minimality of their overall description, since grouping is done as a post-processing step and by using a local-optima-prone clustering algorithm. Moreover, there is not much discussion in their paper on the choice of the number of clusters, nor the format of the final description after the anomalies are clustered. We are not aware of a publicly available implementation of SRF to compare with our proposed method and hence omit it from the experimental evaluation.

Subspace clustering and Outlier detection: There is a long list of work on subspace clustering [3,7,30,44,53] that aim to find high-density clusters in feature subspaces. (See [48,32] for reviews.) Some others are projection based that work in transformed feature spaces [2,41]. However, these are unsupervised methods and their goal is not explaining labeled data, nor they focus on minimal explanations. There is also a long list of subspace-based outlier *detection* methods [23,31,33,45,46,47], however, those do not address the description problem.

Data description and Rare class discovery: Another line of related work is data description [18, 55] and rare class (or category) characterization [21, 20]. The main goal behind all of these work is to explain the data via a hyperball separating rare-class points from normal points. However, all of them assume that the former cluster in a *single* hyperball, and with the exception of [20], search for a *full-dimensional* enclosing hyperball. As such, they do not address the curse-of-dimensionality or identify multiple clusters embedded in different subspaces.

Subgroup discovery and Rule learning: Classification rule learning algorithms have the objective of generating models consisting of a set of rules inducing properties of all the classes of the target variable, while in subgroup discovery the objective is to discover individual rules of interest (See [22] for an overview). The seminal work of rule based learners such as Ripper [9] and CN2 [8] sequentially mine for rules with high accuracy and coverage. More recently, [15] proposes RuleFit, an ensemble learner where the base learner is a rule generated by a decision tree. A regression/classification is setup using the base learners to identify the rules that are important in discriminating the different classes. Few other work in ensemble learning [13, 19] build ensemble trees. While rules are interpretable, they are learnt with an aim to achieve generalization (to unseen data). This is different from our work where we primarily focus on describing the under-represented class (anomalies) without emphasizing the generalization.

SubgroupMiner [26] extends seminal work in subgroup discovery (MIDOS [57], Explora [25]) to handle numerical and categorical attributes. SD [16] proposes an interactive subgroup discovery technique based on the variation of beam search algorithms guided by expert knowledge. Krimp [56] proposes a greedy MDL based approach to mine few frequent item sets describing a dataset. This method can be used on multi-class data to provide descriptions of each class, it is however limited to categorical attributes. Discriminative pattern mining techniques [39] also assume categorical features and aim to extract contrast patterns (item sets) with large support difference across classes.

A key difference between the techniques discussed above and our work is the summarization scheme described in §3.3. Our MDL based encoding scheme leads to a submodular rule selection with theoretical guarantees that the current subgroup discovery or rule learning algorithms do not explore. The improvement of the summarization scheme is evident from our experiments (See Table 6), comparing x-PACS to several rule learners on various interpretability measures.

Explaining black-box classifiers: Approaches such as [14, 28, 42, 51] aim to explain the decision made by a black-box predictor. LIME [51] finds nearest neighbors to single input labeled example to construct a linear interpretable model that is locally faithful to the predictor. Further, authors propose a submodular optimization framework to pick instances that are representative of the predictions of a classifier. Other work [14, 28] explain the model by perturbing the features to quantify the influence on prediction. However, these work do not aim to explain multiple instances collectively, as such they do not handle the summarization problem, which are hence not comparable to our proposed method.

All in all, none of the existing methods provides all of 1) *collective* (as opposed to individual) explanations, 2) explanations for *multiple* anomalous groups, 3) in characterizing *subspaces*, 4) using interpretable *feature rules* that can 5) *discriminate* anomalies from normal points, 6) aiming to *minimize* description length.

6 Conclusion

We considered the problem of explaining given anomalies in high-dimensional datasets in groups. Our key idea is to describe the data by the patterns it contains. We proposed x-PACS for identifying a small number of low-dimensional anomalous patterns that "pack" similar, clustered anomalies and "compress" the data most succinctly. In designing x-PACS, we combined ideas from data mining (bottom-up algorithms with pruning), optimization (nonlinear quadratic discrimination), information theory (data encoding with bits), and theory of algorithms (nonmonotone submodular function maximization). Our notable contributions are listed as follows.

- **A new desiderata** for the anomaly description problem, enlisting five desired properties (D1–D5),
- **A new problem formulation**, for explaining a given set of anomalies *in groups* (D1),
- **Description algorithm** x-PACS, which provides low-dimensional (D2), interpretable (D3), and discriminative (D4) feature rules per anomalous group,
- **A new anomaly encoding scheme**, based on the minimum description length (MDL) principle, that lends itself to efficient optimization to produce **minimal explanations** (D5) with guarantees.

Through experiments on real-world datasets, we showed the effectiveness of x-PACS both in explanation and detection and superiority to competitive baselines. For reproducibility, all of our source code and datasets are publicly released at https://github.com/meghanathmacha/xPACS.

References

1. C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.
2. C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *SIGMOD*, pages 61–72, 1999.
3. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD*, pages 94–105, 1998.
4. F. Angiulli, F. Fassetti, and L. Palopoli. Detecting outlying properties of exceptional objects. *ACM Trans. Database Syst.*, 34(1), 2009.
5. F. Angiulli, F. Fassetti, and L. Palopoli. Discovering characterizations of the behavior of anomalous subpopulations. *IEEE TKDE*, 25(6):1280–1292, 2013.
6. N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. Submodular maximization with cardinality constraints. In *SODA*, pages 1433–1452, 2014.
7. C. H. Cheng, A. W.-C. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *KDD*, pages 84–93, 1999.
8. P. Clark and T. Niblett. The cn2 induction algorithm. *Machine learning*, 3(4):261–283, 1989.

9. W. W. Cohen. Fast effective rule induction. In *Machine Learning Proceedings 1995*, pages 115–123. Elsevier, 1995.

10. X. H. Dang, I. Assent, R. T. Ng, A. Zimek, and E. Schubert. Discriminative features for identifying and interpreting outliers. In *ICDE*, pages 88–99, 2014.

11. X. H. Dang, B. Micenková, I. Assent, and R. T. Ng. Local outlier detection with interpretation. In *ECML/PKDD*, pages 304–320, 2013.

12. V. Dave, S. Guha, and Y. Zhang. Measuring and fingerprinting click-spam in ad networks. In *SIGCOMM*, pages 175–186. ACM, 2012.

13. H. Deng. Interpreting tree ensembles with intrees. *arXiv preprint arXiv:1408.5456*, 2014.

14. R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*, 2017.

15. J. H. Friedman, B. E. Popescu, et al. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

16. D. Gamberger and N. Lavrac. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17:501–527, 2002.

17. S. O. Gharan and J. Vondrak. Submodular maximization by simulated annealing. In *SODA*, pages 1098–1116. SIAM, 2011.

18. N. Görnitz, M. Kloft, and U. Brefeld. Active and semi-supervised data domain description. In *ECML/PKDD*, pages 407–422, 2009.

19. S. Hara and K. Hayashi. Making tree ensembles interpretable. *arXiv preprint arXiv:1606.05390*, 2016.

20. J. He and J. G. Carbonell. Co-selection of features and instances for unsupervised rare category analysis. In *SDM*, pages 525–536, 2010.

21. J. He, H. Tong, and J. G. Carbonell. Rare category characterization. In *ICDM*, pages 226–235, 2010.

22. F. Herrera, C. J. Carmona, P. González, and M. J. Del Jesus. An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525, 2011.

23. F. Keller, E. Müller, and K. Böhm. HiCS: High contrast subspaces for density-based outlier ranking. In *ICDE*, pages 1037–1048, 2012.

24. F. Keller, E. Müller, A. Wixler, and K. Böhm. Flexible and adaptive subspace search for outlier analysis. In *CIKM*, pages 1381–1390. ACM, 2013.

25. W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in knowledge discovery and data mining*, pages 249–271. American Association for Artificial Intelligence, 1996.

26. W. Klösgen and M. May. Census data mining: an application. In *Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Helsinki, Finland*, 2002.

27. E. M. Knorr and R. T. Ng. Finding intensional knowledge of distance-based outliers. In *VLDB*, pages 211–222, 1999.

28. P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

29. M. Kopp, T. Pevný, and M. Holena. Interpreting and clustering outliers with sapling random forests. In *ITAT*, 2014.

30. H.-P. Kriegel, P. Kröger, M. Renz, and S. H. R. Wurst. A generic framework for efficient subspace clustering of high-dimensional data. In *ICDM*, 2005.

31. H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *PAKDD*, pages 831–838, 2009.

32. H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1):1–58, 2009.

33. H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in arbitrarily oriented subspaces. In *ICDM*, pages 379–388, 2012.

34. C.-T. Kuo and I. Davidson. A framework for outlier description using constraint programming. In *AAAI*, pages 1237–1243, 2016.

35. H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Interpretable and explorable approximations of black box models. *CoRR*, abs/1707.01154, 2017.

36. A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *KDD*, pages 157–166, 2005.

37. K. Lee, B. D. Eoff, and J. Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *ICWSM*, 2011.

38. F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *ICDM*, 2008.
39. E. Loekito and J. Bailey. Mining influential attributes that capture class and group contrast behaviour. In *CIKM*, pages 971–980. ACM, 2008.
40. B. Micenková, R. T. Ng, X. H. Dang, and I. Assent. Explaining outliers by subspace separability. In *ICDM*, pages 518–527, 2013.
41. G. Moise, J. Sander, and M. Ester. P3c: A robust projected clustering algorithm. In *ICDM*, pages 414–425, 2006.
42. G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.
43. A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance. What yelp fake review filter might be doing? In *ICWSM*, 2013.
44. E. Müller, I. Assent, S. Günnemann, R. Krieger, and T. Seidl. Relevant subspace clustering: Mining the most interesting non-redundant concepts in high dimensional data. In *ICDM*, pages 377–386. IEEE, 2009.
45. E. Müller, I. Assent, P. I. Sanchez, Y. Mulle, and K. Böhm. Outlier ranking via subspace analysis in multiple views of the data. In *ICDM*, pages 529–538, 2012.
46. E. Müller, I. Assent, U. Steinhausen, and T. Seidl. Outrank: ranking outliers in high dimensional data. In *ICDE Workshops*, pages 600–603, 2008.
47. E. Müller, M. Schiffer, and T. Seidl. Statistical selection of relevant subspace projections for outlier ranking. In *ICDE*, pages 434–445, 2011.
48. L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. 6(1):90–105, 2004.
49. D. Pelleg and A. Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.
50. T. Pevný and M. Kopp. Explaining anomalies with sapling random forests. In *ITAT*, 2014.
51. M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.
52. J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
53. K. Sequeira and M. J. Zaki. Schism: A new approach for interesting subspace mining. In *ICDM*, pages 186–193, 2004.
54. B. W. Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.
55. D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.
56. J. Vreeken, M. Van Leeuwen, and A. Siebes. Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery*, 23(1):169–214, 2011.
57. S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 78–87. Springer, 1997.
58. H. Zhang, Y. Diao, and A. Meliou. EXstream: Explaining anomalies in event stream monitoring. In *EDBT*, pages 156–167, 2017.